4-30-2018

Decentralized Parameter Estimation

Ben Sirb

DECENTRALIZED PARAMETER ESTIMATION

by

BENJAMIN SIRB

Under the Direction of Xiaojing Ye, PhD

ABSTRACT

We develop efficient numerical methods for solving inverse problems in a decentralized setting. First we consider decentralized optimization of a known objective with known gradient, analyzing the convergence of a decentralized consensus algorithm using delayed stochastic gradient information across the network. Each node privately holds a part of the objective function, and nodes collaboratively solve for the consensus optimal solution of the total objective while they can only communicate with their immediate neighbors. In real-world applications, it is often difficult and sometimes impossible to synchronize the nodes,

and therefore they have to use stale gradient information during computations. We show that the iterates generated converge to a consensual optimal solution as long as the random delays are bounded in expectation and a proper diminishing step size policy is employed. Convergence rates of both objective and consensus are derived. Numerical results on a number of synthetic problems and real-world seismic tomography datasets in decentralized sensor networks are presented. We then consider inverse problems in epidemiology where the disease transmission rate (objective) is unknown, looking to develop an efficient decentralized method for its estimation. As the objective is unknown, we first consider the centralized setting, and then extend our work to the decentralized case. We use an SEIR compartmental model and we assume the transmission rate is time-dependent. Using observed incidence case data, we develop a method for estimating disease transmission rate, which may be used to forecast future incidence cases for cyclic disease epidemics. We test the method on synthetic and real-world datasets. We then investigate whether this method may be modified for extension to the decentralized case. We are motivated by the problem in which regions experience an outbreak of a common, cyclic disease epidemic, and we consider whether collaboration can allow for the recovery of a common transmission rate. We investigate whether the common estimate returned by the method produces accurate forecasts of each local regions future incidence cases. The method is tested on a synthetic dataset using both full and partial data for transmission rate recovery.

INDEX WORDS: Decentralized consensus optimization, delayed gradient, stochastic gradient, decentralized networks, inverse problems, epidemiology, regularization, parameter estimation, forecasting.

DECENTRALIZED PARAMETER ESTIMATION

by

BENJAMIN SIRB

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2018

DECENTRALIZED PARAMETER ESTIMATION

by

BENJAMIN SIRB

Committee Chair: Xiaojing Ye

Committee: Alexandra Smirnova

Michael Stewart

Gerardo Chowell

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2018

ACKNOWLEDGEMENTS

This dissertation work would not have been possible without the help and encouragement from my advisor, Dr. Xiaojing Ye. You pushed me to succeed, and you helped me understand what it takes to do mathematical research. You witnessed my good ideas and my silly mistakes, and you helped guide me through it all. I owe much of my current and future academic success to your dedicated support.

Thank you, Dr. Alexandra Smirnova for your enthusiastic guidance and thoughtful feedback on parameter estimation problems. Your help was instrumental to my progress in this area, and it helped me finalize the overarching direction of my work.

Thank you, Dr. Chowell, for sponsoring part of my research on parameter estimation. Your support helped give me some much needed time to work when I needed it the most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PART 1

DECENTRALIZED CONSENSUS ALGORITHM

1.1 Introduction

In this work, we consider a decentralized consensus optimization problem arising from emerging technologies such as distributed machine learning [1, 2, 3, 4], sensor network [5, 6, 7], and smart grid [8, 9]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a network (undirected graph) where $\mathcal{V} = \{1, 2, \ldots, m\}$ is the node (also called agent, processor, or sensor) set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set. Two nodes i and j are called neighbors if $(i, j) \in \mathcal{E}$. The communications between neighbor nodes are bidirectional, meaning that i and j can communicate with each other as long as $(i, j) \in \mathcal{E}$.

In a decentralized sensor network $\mathcal{G}$, individual nodes can acquire, store, and process data about large-sized objects. Each node i collects data and holds objective function $F_i(x; \xi_i)$ privately where $\xi_i \in \Theta$ is random with fixed but unknown probability distribution in domain Θ to model environmental fluctuations such as noise in data acquisition and/or inaccurate estimation of objective function or its gradient. Here $x \in X$ is the unknown (e.g., the seismic image) to be solved, where the domain $X \subset \mathbb{R}^n$ is compact and convex. Furthermore, we assume that $F_i(\cdot; \xi_i)$ is convex for all $\xi_i \in \Theta$ and $i \in \mathcal{V}$, and we define $f_i(x) = \mathbb{E}_{\xi_i}[F_i(x; \xi_i)]$ which is thus convex with respect to $x \in X$. The goal of decentralized consensus optimization is to solve the minimization problem

$$\underset{x \in X}{\text{minimize}}\, f(x), \quad \text{where } f(x) := \sum_{i=1}^{m} f_i(x) \tag{1.1}$$

with the restrictions that $F_i(x; \xi_i)$, and hence $f_i(x)$, are accessible by node i only, and that nodes i and j can communicate only if $(i, j) \in \mathcal{E}$ during the entire computation.

There are a number of practical issues that need to be taken into consideration in solving the real-world decentralized consensus optimization problem (1.1):

- The partial objective F_i (and f_i) is held privately by node i, and transferring F_i to a data fusion center is either infeasible or cost-ineffective due to data privacy, the large size of F_i, and/or limited bandwidth and communication power overhead of sensors. Therefore, the nodes can only communicate their own estimates of $x \in \mathbb{R}^n$ with their neighbors in each iteration of a decentralized consensus algorithm.

- Since it is often difficult and sometimes impossible for the nodes to be fully synchronized, they may not have access to the most up-to-date (stochastic) gradient information during computations. In this case, the node i has to use out-of-date (stochastic) gradient $\nabla F_i(x_i(t-\tau_i(t)); \xi_i(t-\tau_i(t)))$ where $x_i(t)$ is the estimate of x obtained by node i at iteration t, and $\tau_i(t)$ is the level of (possibly random) delay of the gradient information at t.

- The estimates $\{x_i(t)\}$ by the nodes should tend to be consensual as t increases, and the consensual value is a solution of problem (1.1). In this case, there is a guarantee of retrieving a good estimate of x from any surviving node in the network even if some nodes are sabotaged, lost, or run out of power during the computation process.

In this work, we analyze a decentralized consensus algorithm which takes all the factors above into consideration in solving (1.1). We provide comprehensive convergence analysis of the algorithm, including the decay rates of objective function and disagreements between nodes, in terms of iteration number, level of delays, and network structure etc.

1.1.1 Related work

Distributed computing on networks is an emerging technology with extensive applications in modern machine learning [2, 3, 4], sensor networks [5, 6, 10, 11], and big data analysis [12, 13]. There are two types of scenarios in distributed computing: centralized and decentralized. In the centralized scenario, computations are carried out locally by worker

(slave) nodes while computations of certain global variables must eventually be processed by designated master node or at a center of shared memory during each (outer) iteration. A major effort in this scenario has been devoted to update the global variable more effectively using an asynchronous setting in, for example, distributed centralized alternating direction method of multipliers (ADMM) [14, 15, 16, 17, 18]. In the decentralized scenario considered in this paper, the nodes privately hold parts of objective functions and can only communicate with neighbor nodes during computations. In many real-world applications, decentralized computing is particularly useful when a master-worker network setting is either infeasible or not economical, or the data acquisition and computation have to be carried out by individual nodes which then need to collaboratively solve the optimization problem. Decentralized networks are also more robust to node failure and can better address privacy concerns. For more discussions about motivations and advantages of decentralized computing, see, e.g., [19, 20, 21, 22, 23, 24] and references therein.

Decentralized consensus algorithms take the data distribution and communication restriction into consideration, so that they can be implemented at individual nodes in the network. In the *ideal synchronous case* of decentralized consensus where all the nodes are coordinated to finish computation and then start to exchange information with neighbors in each iteration, a number of developments have been made. A class of methods is to rewrite the consensus constraints for minimization problem (1.1) by introducing auxiliary variables between neighbor nodes (i.e., edges), and apply ADMM (possibly with linearization or preconditioning techniques) to derive an implementable decentralized consensus algorithm [25, 26, 27, 28, 29, 30]. Most of these methods require each node to solve a local optimization problem every iteration before communication, and reach a convergence rate of $O(1/T)$ in terms of outer iteration (communication) number T for general convex objective functions $\{f_i\}$. First-order methods based on decentralized gradient descent require less computational cost at individual nodes such that between two communications they only perform one step of a gradient descent-type update at the weighted average of previous iterates obtained from neighbors. In particular, Nesterov's optimal gradient scheme is employed in decentralized

gradient descent with diminishing step sizes to achieve rate of $O(1/T)$ in [19], where an alternative gradient method that requires excessive communications in each inner iteration is also developed and can reach a theoretical convergence rate of $O(\log T/T^2)$, despite that it seems to work less efficiently in terms of communications than the former in practice. A correction technique is developed for decentralized gradient descent with convergence rate as $O(1/T)$ with constant step size in [22], which results in a saddle-point algorithm as pointed out in [31]. In [11], the authors combine Nesterov's gradient scheme and a multiplier-type auxiliary variable to obtain a fast optimality convergence rate of $O(1/T^2)$. Other first-order decentralized methods have also been developed recently, such dual averaging [32]. Additional constraints for primal variables in decentralized consensus optimization (1.1) are considered in [33].

In real-world decentralized computing, it is often difficult and sometimes impossible to coordinate all the nodes in the network such that their computation and communication are perfectly synchronized. One practical approach for such *asynchronous consensus* is using a broadcast scenario where in each (outer) iteration, one node in the network is assumed to wake up at random and broadcasts its value to neighbors (but does not hear them back). A number of algorithms for broadcast consensus are developed, for instance, in [34, 5, 35, 36]. In particular, [36] develops a consensus optimization algorithm for (1.1) in the setting where every iteration one node in the network broadcasts its value to the neighbors, but there are no delays in (sub)gradients during their updates. Another important issue in the asynchronous setting is that nodes may have to use out-of-date (stale) gradient information during updates [20, 37]. This delayed scenario in gradient descent is considered in a distributed but not decentralized setting in [38, 39, 40, 41]. In addition, analysis of stochastic gradient in distributed computing is also carried out in [38, 42]. In [43], linear convergence rate of optimality is derived for strongly convex objective functions with delays. Extending [38], a *fixed* delay at all nodes is considered in dual averaging [44] and gradient descent [45] in a decentralized setting, but they did not consider more practical and useful *random* delays, and there are no convergence rates on node consensus provided in these papers. In [37], both

random delays in communications and gradients are considered, however, no convergence rate is established.

1.1.2 Contributions

The contribution of this work is in three phases. First, we consider a general decentralized consensus algorithm with randomly delayed and stochastic gradient (Section 1.2). In this case, the nodes do not need to be synchronized and they may only have access to stale gradient information. This renders stochastic gradients with random delays at different nodes in their gradient updates, which is suitable for many real-world decentralized computing applications.

Second, we provide a comprehensive convergence analysis of the proposed algorithm (Section 1.3). More precisely, we derive convergence rates for both the objective function (optimality) and disagreement (feasibility constraint of consensus), and show their dependency on the characteristics of the problem, such as Lipschitz constants of (stochastic) gradients and spectral gaps of the underlying network.

Third, we conduct a number of numerical experiments on synthetic and real datasets to validate the performance of the proposed algorithm (Section 1.4). In particular, we examine the convergence on synthetic decentralized least squares, robust least squares, and logistic regression problems. We also present the numerical results on the reconstruction of several seismic images in decentralized wireless sensor networks.

1.1.3 Notations and assumptions

All vectors are column vectors unless otherwise noted. We denote by $x_i(t) \in \mathbb{R}^n$ the estimate of node i at iteration t, and $x(t) = (x_1(t), \ldots, x_m(t))^\top \in \mathbb{R}^{m\times n}$. We denote $\|x\| \equiv \|x\|_2$ if x is a vector and $\|x\| \equiv \|x\|_F$ if x is a matrix, which should be clear by the context. For any two vectors of same dimension, $\langle x, y\rangle$ denotes their inner product, and $\langle x, y\rangle_Q := \langle x, Qy\rangle$ for symmetric positive semidefinite matrix Q. For notational simplicity, we use $\langle x, y\rangle = \sum_{i=1}^m \langle x_i, y_i\rangle$ where x_i and y_i are the i-th row of the $m \times n$ matrices x and y

respectively. Such matrix inner product is also generalized to $\langle x, y\rangle_Q$ for matrices x and y. In this work, we set the domain $X := \{x \in \mathbb{R}^n : \|x\|_\infty \le R\}$ for some $R > 0$, which can be thought of as the maximum pixel intensity in reconstructed images. We further denote $\mathcal{X} := X^m \subset \mathbb{R}^{m\times n}$.

For each node i, we define $f_i(x) := \mathbb{E}_{\xi_i}[F_i(x;\xi_i)]$ as the expectation of objective function, and $g_i(t) := \nabla F_i(x(t);\xi_i(t))$ (here the gradient ∇ is taken with respect to x) is the stochastic gradient at $x_i(t)$ at node i. We let $\tau_i(t)$ be the delay of gradient at node i in iteration t, and $\tau(t) = (\tau_1(t),\dots,\tau_m(t))^\top$. We write $f(x(t))$ in short for $\sum_{i=1}^m f_i(x_i(t)) \in \mathbb{R}$, $x(t-\tau(t))$ for $(x_1(t-\tau_1(t)),\dots,x_m(t-\tau_m(t)))^\top \in \mathbb{R}^{m\times n}$, and $g(t-\tau(t))$ for $(g_1(t-\tau_1(t)),\dots,g_m(t-\tau_m(t)))^\top \in \mathbb{R}^{m\times n}$. We assume f_i is continuously differentiable, ∇f_i has Lipschitz constant L_i, and denote $L := \max_{1\le i\le m} L_i$.

Let $x^* \in \mathbb{R}^n$ be a solution of (1.1), we denote $\mathbf{1}(x^*)^\top$ simply by x^* in this work which is clear by the context, for instance $f(x^*) = f(\mathbf{1}(x^*)^\top) = \sum_{i=1}^m f_i(x^*)$. Furthermore, we let $y(T) := (1/T)\sum_{t=1}^T x(t+1)$ be the running average of $\{x(t+1) : 1 \le t \le T\}$, and $z(T) := (1/m)\sum_{i=1}^m y(T)$ be the consensus average of $y(T)$. We denote $J = (1/m)\mathbf{1}\mathbf{1}^\top$, then $z(T) = Jy(T)$. Note that for all T, $z(T)$ is always consensual but $x(T), y(T)$ may not be.

An important ingredient in decentralized gradient descent is the mixing matrix $W = [w_{ij}]$ in (1.2). For the algorithm to be implementable in practice, $w_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$. We assume that W is symmetric and $\sum_{j=1}^m w_{ij} = 1$ for all i, hence W is doubly stochastic, namely $W\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\top W = \mathbf{1}^\top$ where $\mathbf{1} = (1,\dots,1)^\top \in \mathbb{R}^m$. With the assumption that the network $\mathcal{G}$ is simple and connected, we know $\|W\|_2 = 1$ and eigenvalue 1 of W has multiplicity 1 by the Perron-Frobenius theorem [46]. As a consequence, $Wx = x$ if and only if x is consensual, i.e., $x = c\mathbf{1}$ for some $c \in \mathbb{R}$. We further assume $W \succeq 0$ (otherwise use $\frac{1}{2}(I + W) \succeq 0$ since stochastic matrix W has spectral radius 1). Given a network $\mathcal{G}$, there are different ways to design the mixing matrix W. For some optimal choices of W, see, e.g., [47, 48].

Now we make several assumptions that are necessary in our convergence analysis.

1. The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is undirected, simple, and connected.

2. For all i and x, the stochastic gradient is unbiased, i.e., $\mathbb{E}_{\xi_i}[\nabla F_i(x;\xi_i)] = \nabla f_i(x)$, and $\mathbb{E}_{\xi_i}[\|\nabla F_i(x;\xi_i) - \nabla f_i(x)\|^2] \leq \sigma^2$ for some $\sigma > 0$.

3. The delays $\tau_i(t)$ may follow different distributions at different nodes, but their second moments are assumed to be uniformly bounded, i.e., there exists $B > 0$ such that $\mathbb{E}[\tau_i(t)^2] \leq B^2$ for all $i = 1, \ldots, m$ and iteration t.

Since the domain X is compact and ∇f_i are all Lipschitz continuous, we know $\|\nabla f_i\|$ is uniformly bounded. Furthermore, $\mathbb{E}[\|\nabla F_i(\cdot,\xi_i)\|] \leq \mathbb{E}[\|\nabla F_i(\cdot,\xi_i) - \nabla f_i(\cdot)\|] + \|\nabla f_i(\cdot)\| \leq \sigma + \|\nabla f_i(\cdot)\|$, we know $\mathbb{E}[\|\nabla F_i(\cdot,\xi_i)\|]$ is also uniformly bounded. Therefore, we denote by $G > 0$ the uniform bound such that $\|\nabla f_i\|, \mathbb{E}[\|\nabla F_i(\cdot,\xi_i)\|] \leq G$ for all i. We also assume that the random delay $\tau_i(t)$ and error of inexact gradient $\epsilon_i(t) := g_i(t) - \nabla f_i(x(t))$ are independent.

1.2 Algorithm

Taking the delayed stochastic gradient and the constraint that nodes can only communicate with immediate neighbors, we propose the following decentralized delayed stochastic gradient descent method for solving (1.1). Starting from an initial guess $\{x_i(0) : i = 1, \ldots, m\}$, each node i performs the following updates iteratively:

$$x_i(t+1) = \Pi_X \left[\sum_{j=1}^{m} w_{ij} x_j(t) - \alpha(t) g_i(t - \tau_i(t)) \right]. \tag{1.2}$$

Namely, in each iteration t, the nodes exchange their most recent $x_i(t)$ with their neighbors. Then each node takes weighted average of the received local copies using weights w_{ij} and performs a gradient descent type update using a stochastic gradient $g_i(t - \tau_i(t))$ with delay $\tau_i(t)$ and step size $\alpha(t)$, and projects the result onto X. In addition, each node i tracks its

own running average $y_i(t) = (1/t) \cdot \sum_{s=1}^{t} x_i(s+1)$ by simply updating $y_i(t) = (1-1/t) \cdot y_i(t-1) + (1/t) \cdot x_i(t+1)$ in iteration t.

Following the matrix notation in Section 1.1.3, the iteration (1.2) can be written as

$$x(t+1) = \Pi_{\mathcal{X}}[Wx(t) - \alpha(t)g(t-\tau(t))]. \tag{1.3}$$

Here the projection $\Pi_{\mathcal{X}}$ is accomplished by each node projecting to X due to the definition of X in Section 1.1.3, which does not require any coordination between nodes. Note that the update (1.3) is also equivalent to

$$x(t+1) = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \left\{ \langle g(t-\tau(t)), x \rangle + \frac{1}{2\alpha(t)} \|x - Wx(t)\|^2 \right\}. \tag{1.4}$$

We may refer to the proposed decentralized delayed stochastic gradient descent algorithm by any of (1.2), (1.3), and (1.4) since they are equivalent.

1.3 Convergence Analysis

In this section, we provide a comprehensive convergence analysis of the proposed algorithm (1.4) by employing a proper step size policy. In particular, we derive convergence rates for both of the disagreement (Theorem 3) and objective function value (Theorem 6).

Lemma 1. *For any $x \in \mathbb{R}^{m \times n}$, its projection onto $\mathcal{X}$ yields nonincreasing disagreement. That is*

$$\|(I - J)\,\Pi_{\mathcal{X}}(x)\| \leq \|(I - J)x\|. \tag{1.5}$$

Proof. It suffices to show that for any fixed $R > 0$ and $X = \{x \in \mathbb{R}^m : \|x\|_\infty \le R\}$, there is

$$\|(I - J)\,\Pi_X(x)\| \le \|(I - J)x\| \tag{1.6}$$

for all $x \in \mathbb{R}^m$. Note that for $x = (x_1, x_2, \ldots, x_m)^\top \in \mathbb{R}^m$, there is

$$\|(I - J)x\|^2 = \sum_{i=1}^{m} (x_i - \overline{x})^2$$

where $\overline{x} := (1/m)\sum_{i=1}^m x_i$. We only need to show that if all $\{x_i : x_i < -R\}$ are projected to $-R$ then $\|(I - J)x\|^2$ will reduce. Without loss of generality, suppose $x_1, \ldots, x_\ell < -R$ and $x_{\ell+1}, \ldots, x_m \ge -R$, and let denote the means of these two groups by

$$\mu_1 := \frac{1}{\ell}\sum_{i=1}^{\ell} x_i < -R \quad \text{and} \quad \mu_2 := \frac{1}{m-\ell}\sum_{i=\ell+1}^{m} x_i \ge -R. \tag{1.7}$$

Then we have $\overline{x} = (\ell\mu_1 + (m-\ell)\mu_2)/m$, and

$$\begin{aligned}
&\|(I - J)x\|^2 \\
&= \sum_{i=1}^{m} (x_i - \overline{x})^2 = \sum_{i=1}^{m} (x_i - \frac{\ell\mu_1 + (m-\ell)\mu_2}{m})^2 \\
&= \sum_{i=1}^{\ell} (x_i - \frac{\ell\mu_1 + (m-\ell)\mu_2}{m})^2 + \sum_{i=\ell+1}^{m} (x_i - \frac{\ell\mu_1 + (m-\ell)\mu_2}{m})^2 \\
&= \sum_{i=1}^{\ell} \left((x_i - \mu_1) + \frac{m-\ell}{m}(\mu_1 - \mu_2)\right)^2 + \sum_{i=\ell+1}^{m} \left((x_i - \mu_2) + \frac{\ell}{m}(\mu_2 - \mu_1)\right)^2 \\
&= \sum_{i=1}^{\ell} (x_i - \mu_1)^2 + 2\frac{m-\ell}{m}(\mu_1 - \mu_2)\sum_{i=1}^{\ell}(x_i - \mu_1) + \ell\left(\frac{m-\ell}{m}\right)^2 (\mu_1 - \mu_2)^2 \\
&\quad + \sum_{i=\ell+1}^{m} (x_i - \mu_2)^2 + 2\frac{\ell}{m}(\mu_2 - \mu_1)\sum_{i=\ell+1}^{m}(x_i - \mu_2) + (m-\ell)\left(\frac{\ell}{m}\right)^2 (\mu_2 - \mu_1)^2
\end{aligned} \tag{1.8}$$

After $x_1, \ldots, x_\ell$ are projected to $-R$ (and $x_{\ell+1}, \ldots, x_m$ remain unchanged), their mean is updated from μ_1 to $-R$ for all $i = 1, \ldots, \ell$, and $\mu_2 - \mu_1 (\ge 0)$ reduces to $\mu_2 + R (\ge 0)$. Therefore, the first, third, and sixth terms in the right hand side of (1.8) are decreased,

the second and fifth terms remain zero, and the fourth term remains unchanged. Thus $\|(I-J)x\|$ reduces after projection to $[-R,\infty)^m$. A similar argument implies that projecting $\{x_i : x_i > R\}$ to R will further reduce $\|(I-J)x\|^2$. Therefore projecting x to X, i.e., projecting to $[-R,\infty)^m$ and then $(-\infty,R]^m$, reduces $\|(I-J)x\|^2$. □

Lemma 2. *Let $c_1 \geq 0$ and $c_2 > 0$, and define $\alpha(t) = 1/(c_1 + c_2\sqrt{t})$. Then for any $\lambda \in (0,1)$ there is*

$$\sum_{s=0}^{t-1} \alpha(s)\lambda^{t-s-1} \leq \frac{\sqrt{\pi}\lambda^{-2}}{c_2\sqrt{t}\log(\lambda^{-1})} + O\left(\lambda^t\right) = O\left(\frac{1}{\sqrt{t}}\right) \tag{1.9}$$

for all $t = 1, 2, \ldots$.

Proof. First, we note that

$$\sum_{s=0}^{t-1} \alpha(s)\lambda^{t-1-s} = \alpha(0)\lambda^{t-1} + \alpha(1)\lambda^{t-2} + \sum_{s=2}^{t-1} \alpha(s)\lambda^{t-1-s} \tag{1.10}$$

which means that the rate is upper bounded by the last sum on the right side above since the first two tend to 0 at a linear rate $\lambda \in (0,1)$.

Note that for all $w \in [s-1, s]$ we have $\frac{1}{\sqrt{s}} \leq \frac{1}{\sqrt{w}}$ and $\lambda^{-s} \leq \lambda^{-(w+1)}$ since $\lambda \in (0,1)$, and therefore

$$\alpha(s)\lambda^{t-1-s} = \frac{\lambda^{t-1-s}}{c_1 + c_2\sqrt{s}} \leq \frac{\lambda^{t-1}\lambda^{-s}}{c_2\sqrt{s}} \leq \frac{\lambda^{t-1}\lambda^{-(w+1)}}{c_2\sqrt{w}} = \frac{\lambda^{t-2-w}}{c_2\sqrt{w}}. \tag{1.11}$$

This inequality allows us to bound the last term on right hand side of (1.10) by

$$\sum_{s=2}^{t-1} \alpha(s)\lambda^{t-1-s} \leq \sum_{s=2}^{t-1} \int_{s-1}^{s} \frac{\lambda^{t-2-w}}{c_2\sqrt{w}} dw = \int_1^{t-1} \frac{\lambda^{t-2-w}}{c_2\sqrt{w}} dw = \frac{2\lambda^{t-2}}{c_2} I_t, \tag{1.12}$$

where I_t is defined by

$$I_t := \frac{1}{2}\int_1^{t-1} \frac{\lambda^{-w}}{\sqrt{w}} dw. \tag{1.13}$$

By changing of variable $w = u^2$, we obtain $I_t = \int_1^{\sqrt{t-1}} \lambda^{-u^2} du$. Now we have that

$$\begin{aligned} I_t^2 &= \int_1^{\sqrt{t-1}} \int_1^{\sqrt{t-1}} \lambda^{-(u^2+v^2)}\, dudv = \int_1^{\sqrt{t-1}} \int_1^{\sqrt{t-1}} e^{-(u^2+v^2)\log\lambda}\, dudv \\ &\le \int_0^{\sqrt{t}} \int_0^{\sqrt{t}} e^{-(u^2+v^2)\log\lambda}\, dudv = 2\int_0^{\pi/4} \int_0^{\sqrt{t}/\cos\theta} e^{-\rho^2\log\lambda} \rho\, d\rho d\theta \\ &= -\frac{1}{\log\lambda} \int_0^{\pi/4} (e^{-t\log\lambda/\cos^2(\theta)} - 1)\, d\theta < -\frac{1}{\log\lambda} \int_0^{\pi/4} e^{-t\log\lambda/\cos^2(\theta)}\, d\theta \end{aligned} \tag{1.14}$$

where the third equality comes from changing to a polar system with the substitutions $u = \rho\cos\theta$ and $v = \rho\sin\theta$. Note that $\cos^{-2}(\theta) - (1 + 4\theta/\pi) \le 0$ for all $\theta \in [0, \pi/4]$ since $\cos^{-2}(\theta) - 1 - 4\theta/\pi$ is convex with respect to θ and vanishes at $\theta = 0$ and $\theta = \pi/4$. Therefore

$$I_t^2 \le -\frac{1}{\log\lambda} \int_0^{\pi/4} e^{-t\log\lambda(1+4\theta/\pi)} d\theta \le \frac{\pi\lambda^{-2t}}{4t(\log\lambda)^2}. \tag{1.15}$$

Hence the sum in (1.12) is bounded by

$$\sum_{s=2}^{t-1} \alpha(s)\lambda^{t-1-s} \le \frac{2\lambda^{t-2}}{c_2} I_t \le \frac{2\lambda^{t-2}}{c_2} \frac{\sqrt{\pi}\lambda^{-t}}{2\sqrt{t}\log(\lambda^{-1})} = \frac{\sqrt{\pi}\lambda^{-2}}{c_2\sqrt{t}\log(\lambda^{-1})} \tag{1.16}$$

which completes the proof. □

Now we are ready to prove the convergence rate of disagreement in $x(t)$ and $y(t)$. In particular, we show that $(\sum_{i=1}^m \|x_i(t) - \bar{x}(t)\|^2)^{1/2}$ decays at the rate of $O(1/\sqrt{t})$, where $\bar{x}(t) = (1/m)\sum_{i=1}^m x_i(t)$. The same convergence rate holds for the disagreement of running average $y(t)$. More specifically, these convergence rates are given by the bounds in the following theorem.

Theorem 3. *Let $\{x(t)\}$ be the iterates generated by Algorithm (1.4) with $\alpha(t) = [2(L + \eta\sqrt{t})]^{-1}$ for some $\eta > 0$, and $\lambda = \|W - J\|$. Then λ is the second largest eigenvalue of W and hence $\lambda \in (0, 1)$. Moreover, the disagreement of $x(t)$ is bounded by*

$$\mathbb{E}[\|(I - J)x(t)\|] \leq \sqrt{m}G \sum_{s=0}^{t-1} \alpha(s)\lambda^{t-s-1} \leq \frac{\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{t}\log(\lambda^{-1})} = O\left(\frac{1}{\sqrt{t}}\right), \tag{1.17}$$

and the disagreement of running average $y(T) = (1/m)\sum_{t=1}^{T} x(t+1)$ is bounded by

$$\mathbb{E}[\|(I - J)y(T)\|] \leq \frac{2\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{T}\log(\lambda^{-1})} = O\left(\frac{1}{\sqrt{T}}\right). \tag{1.18}$$

Proof. We first prove the bound on disagreement between $\{x_i(t) : 1 \leq i \leq m\}$, i.e., (1.17), by induction. It is trivial to show that this bound holds for $t = 1$. Assuming (1.17) holds for t, we have

$$\begin{aligned}
\mathbb{E}[\|(I - J)x(t+1)\|] &= \mathbb{E}[\|(I - J)\,\Pi_{\mathcal{X}}(Wx(t) - \alpha(t)g(t - \tau(t)))\|] \\
&\leq \mathbb{E}[\|(I - J)(Wx(t) - \alpha(t)g(t - \tau(t)))\|] \\
&\leq \mathbb{E}[\|(I - J)Wx(t)\|] + \alpha(t)\,\mathbb{E}[\|(I - J)g(t - \tau(t))\|] \\
&\leq \mathbb{E}[\|(I - J)Wx(t)\|] + \alpha(t)\sqrt{m}G
\end{aligned} \tag{1.19}$$

where we used Lemma 1 in the first inequality, and $\|I - J\| \leq 1$ and $\mathbb{E}[\|g_i(t - \tau_i(t))\|] \leq G$ in the last inequality. Noting that $J^2 = J$ and $JW = WJ = J$, we have $(W - J)(I - J) = (I - J)W$.

Therefore, we obtain

$$\begin{aligned}\mathbb{E}[\|(I-J)x(t+1)\|] \leq & \mathbb{E}[\|(I-J)Wx(t)\|] + \alpha(t)\sqrt{m}G \\ = & \mathbb{E}[\|(W-J)(I-J)x(t)\|] + \alpha(t)\sqrt{m}G \\ \leq & \mathbb{E}[\|(W-J)\|\|(I-J)x(t)\|] + \alpha(t)\sqrt{m}G \\ \leq & \lambda\sqrt{m}G\sum_{s=0}^{t-1}\alpha(s)\lambda^{t-s-1} + \alpha(t)\sqrt{m}G \\ = & \sqrt{m}G\sum_{s=0}^{t}\alpha(s)\lambda^{t-s}\end{aligned} \tag{1.20}$$

where we used the induction assumption for t in the last inequality. Applying Lemma 2 to the bound yields the second inequality in (1.17), which shows that $\mathbb{E}[\|(I-J)x(t)\|]$ decays at rate $O(1/\sqrt{t})$.

By convexity of $\|\cdot\|$ and definition of $y(T)$, we obtain that

$$\mathbb{E}[\|(I-J)y(T)\|] \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|(I-J)x(t+1)\|] \leq \frac{2\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{T}\log(\lambda^{-1})} \tag{1.21}$$

by applying (1.17) and using $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$. Therefore the disagreement $\mathbb{E}[\|(I-J)y(T)\|]$ also decays at rate of $O(1/\sqrt{T})$. □

The convergence rate of disagreement also yields an estimate of differences between consecutive iterates $x(t)$ and $x(t+1)$, which is given by the following corollary.

Corollary 1. *Let $\{x(t)\}$ be the iterates generated by Algorithm (1.4) with the settings of $\alpha(t)$, λ, and η same as in Theorem 3. Then there is*

$$\mathbb{E}[\|x(t+1)-x(t)\|] \leq \frac{C}{\sqrt{t}}, \tag{1.22}$$

where $C := \frac{\sqrt{m}G}{\eta}\left[\frac{\sqrt{\pi}\lambda^{-2}}{\log(\lambda^{-1})} + \frac{1}{2}\right]$ is a constant independent of t.

Proof. According to the update (1.4) or equivalently (1.3), we have

$$\begin{aligned}\mathbb{E}[\|x(t+1)-x(t)\|] &= \mathbb{E}[\|\,\Pi_{\mathcal{X}}[Wx(t)-\alpha(t)g(t-\tau(t))]-x(t)\|] \\ &\le \mathbb{E}[\|(I-W)x(t)+\alpha(t)g(t-\tau(t)))\|] \\ &\le \mathbb{E}[\|(I-W)x(t)\|]+\alpha(t)\,\mathbb{E}[\|g(t-\tau(t)))\|]\end{aligned} \tag{1.23}$$

where we used the facts that $x(t)\in\mathcal{X}$ and that projection $\Pi_{\mathcal{X}}$ is non-expansive in the first inequality. Note that $WJ=J$ and hence $I-W=(I-W)(I-J)$, we have

$$\mathbb{E}[\|(I-W)x(t)\|]=\mathbb{E}[\|(I-W)(I-J)x(t)\|]\le\mathbb{E}[\|(I-J)x(t)\|]\le\frac{\sqrt{\pi m}G\lambda^{-2}}{\eta\sqrt{t}\log(\lambda^{-1})}$$

where we used the fact that $\|I-W\|\le 1$ in the first inequality and applied Theorem 3 to obtain the second inequality. Furthermore, we have by the definition of $\alpha(t)$ that

$$\|\alpha(t)g(t-\tau(t))\|\le\sqrt{m}\alpha(t)G=\frac{\sqrt{m}G}{2(L+\eta\sqrt{t})}\le\frac{\sqrt{m}G}{2\eta\sqrt{t}}. \tag{1.24}$$

Applying the two inequalities above to (1.23) yields (1.22). □

From the estimate of difference between consecutive iterates, we can also bound the expected difference between $x(t)$ and $x(t-\tau(t))$ as follows.

Corollary 2. *Let $\{x(t)\}$ be the iterates generated by Algorithm* (1.4) *with the settings of $\alpha(t)$, λ, and η same as in Theorem 3. Then there is*

$$\mathbb{E}[\|x(t)-x(t-\tau(t))\|]\le C\left(\frac{\sqrt{2m}B}{\sqrt{t}}+\frac{4mB^2}{t}\right)=O\left(\frac{1}{\sqrt{t}}\right) \tag{1.25}$$

where C is the constant defined in Corollary 1. In particular, if $t\ge 8mB^2$, there is $\mathbb{E}[\|x(t)-x(t-\tau(t))\|]\le\frac{2\sqrt{2m}CB}{\sqrt{t}}$.

Proof. We first define $\bar{\tau}(t):=\max\{\tau_i(t):1\le i\le m\}$. Then there is $\mathbb{E}[|\bar{\tau}(t)|^2]\le\mathbb{E}[\sum_{i=1}^m|\tau_i(t)|^2]\le mB^2$. Without loss of generality, we assume that $0\le\bar{\tau}(t)\le t-2$

for every given t, i.e., we consider the convergence rate when every node has successfully computed their own gradient at least twice. Then we obtain that

$$\begin{aligned}
&\mathbb{E}[\|x(t)-x(t-\tau(t))\|] \\
&\le \mathbb{E}\left[\sum_{s=1}^{\bar{\tau}(t)}\|x(t-s+1)-x(t-s)\|\right] \le C\,\mathbb{E}\left[\sum_{s=1}^{\bar{\tau}(t)}\frac{1}{\sqrt{t-s}}\right] \\
&= C\,\mathbb{E}\left[\sum_{s=t-\bar{\tau}(t)}^{t-1}\frac{1}{\sqrt{s}}\right] \le C\,\mathbb{E}\left[\int_{t-\bar{\tau}(t)-1}^{t-1}\frac{1}{\sqrt{s}}ds\right] \\
&= 2C\,\mathbb{E}\left[\sqrt{t-1}-\sqrt{t-\bar{\tau}(t)-1}\right] \le 2C\,\mathbb{E}\left[\frac{\bar{\tau}(t)}{\sqrt{t-1}+\sqrt{t-\bar{\tau}(t)-1}}\right] \\
&\le C\,\mathbb{E}\left[\frac{\bar{\tau}(t)}{\sqrt{t-\bar{\tau}(t)-1}}\right]
\end{aligned} \tag{1.26}$$

where we used triangle inequality to obtain the first inequality, applied Corollary 1 to obtain the second inequality, and used the fact that $\bar{\tau}(t)\ge 0$ to obtain the last inequality above. Note that there is

$$\begin{aligned}
\mathbb{E}\left[\frac{\bar{\tau}(t)}{\sqrt{t-\bar{\tau}(t)-1}}\right] &= \sum_{s=0}^{\lfloor t/2\rfloor-1}\frac{s}{\sqrt{t-s-1}}\,\mathrm{P}(\bar{\tau}(t)=s)+\sum_{s=\lfloor t/2\rfloor}^{t-2}\frac{s}{\sqrt{t-s-1}}\,\mathrm{P}(\bar{\tau}(t)=s) \\
&\le \frac{\sqrt{2}}{\sqrt{t}}\sum_{s<t/2}s\,\mathrm{P}(\bar{\tau}(t)=s)+(t-2)\sum_{s\ge t/2}\mathrm{P}(\bar{\tau}(t)=s) \\
&\le \frac{\sqrt{2m}B}{\sqrt{t}}+\frac{4mB^2(t-2)}{t^2} \le \frac{\sqrt{2m}B}{\sqrt{t}}+\frac{4mB^2}{t} = O\left(\frac{1}{\sqrt{t}}\right)
\end{aligned} \tag{1.27}$$

where we used the fact that $\sqrt{t-s-1}\ge\sqrt{t/2}$ if $0\le s\le\lfloor t/2\rfloor-1$ and $s/\sqrt{t-s-1}\le t-2$ if $\lfloor t/2\rfloor\le s\le t-2$ to obtain the first inequality, and $\sum_{s<t/2}s\,\mathrm{P}(\bar{\tau}(t)=s)\le\mathbb{E}[\bar{\tau}(t)]\le\sqrt{\mathbb{E}[\bar{\tau}(t)^2]}=\sqrt{m}B$ and $\sum_{s\ge t/2}\mathrm{P}(\bar{\tau}(t)=s)=\mathrm{P}(\bar{\tau}(t)\ge t/2)\le(4/t^2)\,\mathbb{E}[\bar{\tau}(t)^2]\le 4mB^2/t^2$ (by Chebyshev's inequality) in the second inequality.

In particular, it is easy to verify that when $t\ge 8mB^2$, it follows that $\sqrt{2m}B/\sqrt{t}\ge 4mB^2/t$ and hence $\mathbb{E}\left[\frac{\bar{\tau}(t)}{\sqrt{t-\bar{\tau}(t)-1}}\right]\le\frac{2\sqrt{2m}B}{\sqrt{t}}$. Combining (1.26) and (1.27) completes the proof. □

Without loss of generality and for sake of notation simplicity, we assume iteration number $t > 8mB^2$ and $\mathbb{E}[\|x(t)-x(t-\tau(t))\|] \leq \frac{2\sqrt{2m}CB}{\sqrt{t}}$ in the remaining derivations. The decay rate $O(1/\sqrt{t})$ of $\mathbb{E}[\|x(t)-x(t-\tau(t))\|]$ is useful to estimate the convergence rate of objective function value later.

Lemma 4. *Let $\{x(t)\}$ be the iterates generated by Algorithm (1.3), then the following inequality holds for all $T \geq 1$:*

$$\sum_{t=1}^{T} \mathbb{E}\left\langle \nabla f(x(t)) - \nabla f(x(t-\tau(t))), x(t+1) - x^* \right\rangle \leq 8\sqrt{2nLT}mRCB \tag{1.28}$$

where C is the constant defined in Corollary 1.

Proof. By the Cauchy-Schwarz inequality,

$$\begin{aligned} &\sum_{t=1}^{T} \langle \nabla f(x(t)) - \nabla f(x(t-\tau(t))), x(t+1) - x^* \rangle \\ \leq &\sum_{t=1}^{T} \|\nabla f(x(t)) - \nabla f(x(t-\tau(t))\| \|x(t+1) - x^*\|. \end{aligned}$$

Note that $\|x(t+1)-x^*\|^2 = \sum_{i=1}^{m} \|x_i(t+1)-x^*\|^2 \leq mn(2R)^2$ due to the bound of $X = \{x \in \mathbb{R}^n : \|x\|_\infty \leq R\}$, and $\mathbb{E}\,\|\nabla f(x(t)) - \nabla f(x(t-\tau(t))\|^2 = \mathbb{E}\left(\sum_{i=1}^{m} \|\nabla f_i(x_i(t)) - \nabla f_i(x_i(t-\tau(t))\|^2 \right) \leq \mathbb{E}\left(\sum_{i=1}^{m} L_i \|x_i(t) - x_i(t-\tau(t))\|^2 \right) \leq L\,\mathbb{E}\,\|x(t) - x(t-\tau(t))\|^2 \leq 2\sqrt{2m}CB/\sqrt{t}$ due to Corollary 2.

Therefore, we obtain

$$\mathbb{E}\left(\sum_{t=1}^{T} \langle \nabla f(x(t)) - \nabla f(x(t-\tau(t))), x(t+1) - x^* \rangle \right) \leq 8\sqrt{2nLT}mRCB$$

by using the fact that $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$.

This completes the proof. □

Now we are ready to prove the convergence rate of objective function value. We first present the estimate of this rate for running averages $y(t)$ in the following theorem.

Theorem 5. *Let $\{x(t)\}$ be the iterates generated by Algorithm (1.3) with $\alpha(t) = [2(L + \eta\sqrt{t})]^{-1}$ for some $\eta > 0$, then*

$$\mathbb{E}[f(y(T))] - f(x^*) \leq \frac{L\mathcal{D}_{\mathcal{X}}^2}{T} + \frac{K}{\sqrt{T}} = O\left(\frac{1}{\sqrt{T}}\right) \tag{1.29}$$

where $y(T) = (1/T)\sum_{t=1}^{T} x(t+1)$ is the running average of $\{x(t)\}$, $\mathcal{D}_{\mathcal{X}} = 2\sqrt{mn}R$ is the diameter of $\mathcal{X}$, and $K := \eta\mathcal{D}_{\mathcal{X}}^2 + 4\sqrt{2mL}\mathcal{D}_{\mathcal{X}}CB + (4m\sigma^2/\eta)$.

Proof. We first note that there is

$$\begin{aligned}
& f(x(t+1)) - f(x^*) = \sum_{i=1}^{m} \big(f_i(x_i(t+1)) - f_i(x^*)\big) \\
= & \sum_{i=1}^{m} \big[f_i(x_i(t+1)) - f_i(x_i(t)) + f_i(x_i(t)) - f_i(x^*)\big] \\
\leq & \sum_{i=1}^{m} \Big[\big\langle \nabla f_i(x_i(t)), x_i(t+1) - x_i(t)\big\rangle + \frac{L_i}{2}\|x_i(t+1) - x_i(t)\|^2 \\
& \qquad + \big\langle \nabla f_i(x_i(t)), x_i(t) - x^*\big\rangle \Big] \\
\leq & \sum_{i=1}^{m} \left[\big\langle \nabla f_i(x_i(t)), x_i(t+1) - x^*\big\rangle + \frac{L_i}{2}\|x_i(t+1) - x_i(t)\|^2 \right] \\
\leq & \big\langle \nabla f(x(t)), x(t+1) - x^*\big\rangle + \frac{L}{2}\|x(t+1) - x(t)\|^2 \\
= & \big\langle g(t-\tau(t)), x(t+1) - x^*\big\rangle + \big\langle \nabla f(x(t)) - g(t-\tau(t)), x(t+1) - x^*\big\rangle \\
& \qquad + \frac{L}{2}\|x(t+1) - x(t)\|^2
\end{aligned} \tag{1.30}$$

where we used the L_i-Lipschitz continuity of ∇f_i and convexity of f_i to obtain the first inequality. Note that $x(t+1)$ is obtained by (1.4) as

$$\begin{aligned}
x(t+1) = & \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \langle g(t-\tau(t)), x\rangle + \frac{1}{2\alpha(t)}\|x - Wx(t)\|^2 \right\} \\
= & \operatorname*{argmin}_{x\in\mathcal{X}} \left\{ \left\langle g(t-\tau(t)) + \frac{1}{\alpha(t)}(I-W)x(t), x \right\rangle + \frac{1}{2\alpha(t)}\|x - x(t)\|^2 \right\}
\end{aligned} \tag{1.31}$$

The optimality of $x(t+1)$ and strong convexity of the objective function in (1.4) imply that

$$\begin{aligned}
&\left\langle g(t-\tau(t)), x(t+1)-x^*\right\rangle \\
\leq &-\frac{1}{\alpha(t)}\left\langle (I-W)x(t), x(t+1)-x^*\right\rangle \\
&+\frac{1}{2\alpha(t)}\left[\|x^*-x(t)\|^2-\|x(t+1)-x(t)\|^2-\|x^*-x(t+1)\|^2\right].
\end{aligned} \tag{1.32}$$

Furthermore, we note that $\mathbf{1}\in \mathrm{Null}(I-W)$ and x^* is consensual, so

$$\begin{aligned}
&-\frac{1}{\alpha(t)}\langle (I-W)x(t), x(t+1)-x^*\rangle \\
=\ &-\frac{1}{\alpha(t)}\langle (I-W)(x(t)-x^*), x(t+1)-x^*\rangle \\
=\ &\frac{1}{2\alpha(t)}\left(\|x(t)-x(t+1)\|_{I-W}^2-\|x(t)-x^*\|_{I-W}^2-\|x(t+1)-x^*\|_{I-W}^2\right) \\
\leq\ &\frac{1}{4\alpha(t)}\|x(t)-x(t+1)\|_{I-W}^2
\end{aligned} \tag{1.33}$$

where we have used the fact that

$$\|x(t)-x(t+1)\|_{I-W}^2\leq 2(\|x(t)-x^*\|_{I-W}^2+\|x(t+1)-x^*\|_{I-W}^2)$$

to obtain the inequality above. We also have that

$$\|x(t)-x(t+1)\|_{I-W}^2\leq \|x(t)-x(t+1)\|^2$$

with which we can further bound (1.33) as

$$-\frac{1}{\alpha(t)}\langle (I-W)x(t), x(t+1)-x^*\rangle\leq \frac{1}{4\alpha(t)}\|x(t)-x(t+1)\|^2.$$

Now applying the inequality above and (1.32) to (1.30), we sum t from 1 to T to get

$$\begin{aligned}\sum_{t=1}^{T} f(x(t+1)) - Tf(x^*) \leq & \sum_{t=1}^{T} \frac{1}{2\alpha(t)} \left(\|x(t)-x^*\|^2 - \|x(t+1)-x^*\|^2\right) \\ & + \sum_{t=1}^{T} \left(\frac{L}{2} - \frac{1}{4\alpha(t)}\right) \|x(t)-x(t+1)\|^2 \\ & + \sum_{t=1}^{T} \left\langle \nabla f(x(t)) - g(t-\tau(t)), x(t+1)-x^* \right\rangle .\end{aligned} \tag{1.34}$$

Note that the running average $y(T) = (1/T)\sum_{t=1}^{T} x(t+1)$ satisfies $f(y(T)) \leq (1/T)\sum_{t=1}^{T} f(x(t+1))$ due to the convexity of all f_i. Therefore, together with (1.34) and the definition of $\alpha(t)$, we have

$$\begin{aligned}& T[f(y(T)) - f(x^*)] \\ \leq & \sum_{t=1}^{T} \left[\frac{1}{2\alpha(t)} \left(\|x(t)-x^*\|^2 - \|x(t+1)-x^*\|^2\right) - \frac{\eta\sqrt{t}}{2}\|x(t)-x(t+1)\|^2\right] \\ & + \sum_{t=1}^{T} \left\langle \nabla f(x(t)) - g(x(t-\tau(t))), x(t+1)-x^* \right\rangle .\end{aligned} \tag{1.35}$$

Now, by taking expectation on both sides of (1.35), we obtain

$$\begin{aligned}T\,\mathbb{E}[f(y(T)) - f(x^*)] \leq & \sum_{t=1}^{T} \left[\frac{1}{2\alpha(t)} \left(e(t) - e(t+1)\right) - \frac{\eta\sqrt{t}}{2}\,\mathbb{E}[\|x(t)-x(t+1)\|^2]\right] \\ & + 8\sqrt{2nLT}mRCB \\ & + \sum_{t=1}^{T} \mathbb{E}\left\langle \nabla f(x(t-\tau(t))) - g(t-\tau(t)), x(t+1)-x^* \right\rangle\end{aligned} \tag{1.36}$$

where we denoted $e(t) := \mathbb{E}[\|x(t)-x^*\|^2]$ for notational simplicity.

Now we work on the last sum of inner products on the right side of (1.36). First we observe that

$$\begin{aligned}
&\mathbb{E}\left\langle \nabla f(x(t-\tau(t))) - g(t-\tau(t)), x(t+1) - x^* \right\rangle \\
=\ &\mathbb{E}\left\langle \nabla f(x(t-\tau(t))) - g(t-\tau(t)), x(t-\tau(t)) - x^* \right\rangle \\
&\quad + \mathbb{E}\left\langle \nabla f(x(t-\tau(t))) - g(t-\tau(t)), x(t+1) - x(t-\tau(t)) \right\rangle.
\end{aligned} \tag{1.37}$$

Note that $g_i(t-\tau_i(t))$ is the stochastic gradient of node i evaluated at iteration $t-\tau_i(t)$, and the stochastic error $g_i(t-\tau_i(t)) - \nabla f_i(x_i(t-\tau_i(t)))$ is independent of $x_i(t-\tau_i(t))$. Therefore, we have

$$\begin{aligned}
&\mathbb{E}\left\langle \nabla f(x(t-\tau(t))) - g(t-\tau(t)), x(t-\tau(t)) - x^* \right\rangle \\
=\ &\sum_{i=1}^{m} \mathbb{E}\left\langle \nabla f_i(x_i(t-\tau_i(t))) - g_i(t-\tau_i(t)), x_i(t-\tau_i(t)) - x^* \right\rangle = 0,
\end{aligned} \tag{1.38}$$

since the stochastic gradients are unbiased. Furthermore, by Young's inequality, we have

$$\begin{aligned}
&\mathbb{E}\left\langle \nabla f(x(t-\tau(t))) - g(t-\tau(t)), x(t+1) - x(t-\tau(t)) \right\rangle \\
\le\ &\frac{2}{\eta\sqrt{t}}\,\mathbb{E}[\|\nabla f(x(t-\tau(t))) - g(t-\tau(t))\|^2] + \frac{\eta\sqrt{t}}{2}\,\mathbb{E}[\|x(t+1) - x(t)\|^2] \\
\le\ &\frac{2m\sigma^2}{\eta\sqrt{t}} + \frac{\eta\sqrt{t}}{2}\,\mathbb{E}[\|x(t+1) - x(t)\|^2]
\end{aligned} \tag{1.39}$$

where we used the fact that $\mathbb{E}[\|\nabla f(x(t-\tau(t))) - g(t-\tau(t))\|^2] \le m\sigma^2$ for all t. Now applying (1.37), (1.38) and (1.39) in (1.36), we have

$$\begin{aligned}
&T\,\mathbb{E}\left[f(y(T)) - f(x^*)\right] \\
\le\ &\sum_{t=1}^{T} \frac{1}{2\alpha(t)}\left(e(t) - e(t+1)\right) + 8\sqrt{2nLT}mRCB + \sum_{t=1}^{T}\frac{2m\sigma^2}{\eta\sqrt{t}} \\
\le\ &\frac{e(1)}{2\alpha(1)} + \sum_{t=2}^{T}\frac{e(t)}{2}\left(\frac{1}{\alpha(t)} - \frac{1}{\alpha(t-1)}\right) + 8\sqrt{2nLT}mRCB + \sum_{t=1}^{T}\frac{2m\sigma^2}{\eta\sqrt{t}}
\end{aligned} \tag{1.40}$$

where we note that $\alpha(t)$ is nonincreasing, and hence $\frac{1}{\alpha(t)} - \frac{1}{\alpha(t-1)} \geq 0$ and

$$\sum_{t=2}^{T} \frac{e(t)}{2} \left(\frac{1}{\alpha(t)} - \frac{1}{\alpha(t-1)} \right) \leq \frac{\mathcal{D}_{\mathcal{X}}^2}{2} \sum_{t=2}^{T} \left(\frac{1}{\alpha(t)} - \frac{1}{\alpha(t-1)} \right) = \frac{\mathcal{D}_{\mathcal{X}}^2}{2} \left(\frac{1}{\alpha(T)} - \frac{1}{\alpha(1)} \right)$$

where we used the fact that $e(t) = \mathbb{E}[\|x(t) - x^*\|^2] \leq \mathcal{D}_{\mathcal{X}}^2 := 4mnR^2$ for all t. Plugging this into (1.40), dividing both sides by T, and using the fact that $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$, we obtain (1.29). This completes the proof. □

We have shown that the running average $y(T)$ makes the objective function decay as in (1.29). However, since each node i obtains its own $y_i(T)$ which may not be consensual (and the left hand side of (1.29) could be negative), we need to look at their consensus average $z(T) = (1/m) \sum_{i=1}^{m} y_i(T)$ and the convergence rate of its objective function value. This is given in the following theorem.

Theorem 6. *Let $x(t)$ be generated by Algorithm (1.2) with $\alpha(t) = [2(L + \eta\sqrt{t})]^{-1}$ for some $\eta > 0$. Let $y(T) = (1/T) \sum_{t=1}^{T} x(t+1)$ be the running average of $x(t)$ and $z(T) = Jy(T) = (1/m) \sum_{i=1}^{m} y_i(T)$ be the consensus average of $y(T)$, then*

$$0 \leq \mathbb{E}[f(z(T))] - f(x^*) \leq \frac{L\mathcal{D}_{\mathcal{X}}^2 + 2\sqrt{m}LC^2}{T} + \frac{K + 2\sqrt{m}CG}{\sqrt{T}} = O\left(\frac{1}{\sqrt{T}} \right) \tag{1.41}$$

where C is defined as in Corollary 1, and $\mathcal{D}_{\mathcal{X}}$ and K are defined as in Theorem 5.

Proof. We first bound the difference between the function values at the running average $y(T)$ and the consensus average $z(T) = Jy(T)$:

$$\begin{aligned} & f(y(T)) - f(z(T)) = \sum_{i=1}^{m} (f_i(y_i(T)) - f_i(z(T))) \\ & \leq \sum_{i=1}^{m} \langle \nabla f_i(z(T)), y_i(T) - z(T) \rangle + \frac{L_i}{2} \|y_i(T) - z(T)\|^2 \\ & \leq \sqrt{m} G \|(I - J)y(T)\| + \frac{L}{2} \|(I - J)y(T)\|^2 \leq \frac{2\sqrt{m}CG}{\sqrt{T}} + \frac{2C^2 L}{T}, \end{aligned} \tag{1.42}$$

where we used convexity of f_i and Lipschitz continuity of ∇f_i in the first inequality, $\|\nabla f_i\| \leq$

G and convexity of $\|\cdot\|^2$ in the second inequality, and Theorem 3 to get the last inequality. Therefore, combining (1.42) and (1.29) from Theorem 5, we obtain the bound in (1.41). Note that $z(T)$ is consensus, so $f(z(T)) \geq f(x^*)$ since x^* is a consensus optimal solution of (1.1). This completes the proof. □

In summary, we have showed that the running average $y_i(T)$, which can be easily updated by each node i, yields convergence in optimality and consensus feasibility. More precisely, Theorem 3 implies that $\|y_i(T) - z(T)\|$ converges to 0 at rate $O(1/\sqrt{T})$ for all nodes i where $z(T) = (1/m)\sum_{i=1}^{m} y_i(T)$ is their consensus average, and Theorem 6 implies that $f(z(T))$ converges to $f(x^*)$ at rate of $O(1/\sqrt{T})$. It is known that $O(1/\sqrt{T})$ is the optimal rate for stochastic gradient algorithms in centralized setting, and hence these two Theorems suggest an encouraging fact that such rate can be retained even if the problem becomes much more complicated, i.e., the gradients are stochastic and delayed, and the computation is carried out in decentralized setting. To retain convergence in this complex setting, we employed a diminishing step size policy as commonly used in stochastic optimization. Such step size policy results in a convergence rate of $O(1/\sqrt{T})$ even without delays and randomness in gradients. Furthermore, due to errors and uncertainties in delayed and stochastic gradients, the iterates may be directed further apart from solution during computations. As a consequence, the constant in the estimated convergence rate appears to depend on the bound of set X rather than the distance between initial guess and solution set as in the setting with non-delayed and non-stochastic gradients.

1.4 Numerical Experiments

In this section, we test algorithm (1.2) on decentralized consensus optimization problem (1.1) with delayed stochastic gradients using a number of synthetic and real datasets. The structure of network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and objective function in (1.1) are explained for each dataset, followed by performance evaluation shown in plots of objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m} \|y_i(T) - z(T)\|^2$ versus the iteration number T, where $y_i(T) = (1/T)\sum_{t=1}^{T} x_i(t+1)$

is the running average of $x_i(t)$ in algorithm (1.2) at each node i, and $z(T) = (1/m)\sum_{i=1}^m y_i(T)$ is the consensus average at iteration T.

1.4.1 Test on synthetic data

We first test on three different types of objective functions using synthetic datasets. In particular, we apply algorithm (1.2) to decentralized least squares, decentralized robust least squares, and decentralized logistic regression problems with different delay and stochastic error combinations. Then we compare the performance of the algorithm with and without delays and stochastic errors in gradients. The performance of the algorithm on different network size m and time comparison with synchronous algorithm are also presented.

In the first set of tests on three different objective functions, we simulate a network of regular 5×5 2-dimensional (2D) lattice of size $m = 25$. We set dimension of unknown x to $n = 10$ and generate an $\hat{x} \in \mathbb{R}^n$ using MATLAB built-in function `rand`, and set the ℓ_∞ radius of X to $R = 1$. For each node i, we generate matrices $A_i \in \mathbb{R}^{p_i \times n}$ with $p_i = 5$ using `randn`, and normalize each column into unit ℓ_2 ball in $\mathbb{R}^{p_i}$ for $i = 1, \ldots, m$. Then we simulate $b_i = A_i\hat{x} + \epsilon_i$ where ϵ_i is generated by `randn` with mean 0 and standard deviation 0.001. For decentralized least squares problem, we set the objective function to $f_i(x) = (1/2)\|A_i x - b_i\|^2$ at node i. Therefore the Lipschitz constant of ∇f_i is $L_i = \|A_i^\top A_i\|_2$, and we further set $L = \max_{1\le i\le m}\{L_i\}$. The initial guess $x_i(0)$ is set to 0 for all i. For each iteration t, the delay $\tau_i(t)$ at each node i is uniformly drawn from integers 1 to B with $B = 5$, 10 and 20. For given t, the stochastic gradient is simulated by setting $\nabla F_i(x_i(t); \xi_i(t)) = A_i^\top(A_i x_i(t) - b_i) + \xi_i(t)$ where $\xi_i(t)$ is generated by `randn` with mean 0 and standard deviation σ set to 0.01 and 0.05. We run our algorithm using step size $\alpha(t) = 1/(2L + 2\eta\sqrt{t})$ with $\eta = 0.01$. The objective function $f(z(T)) - f^*$ and disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ versus the iteration number T are plotted in the top row of Figure 1.1, where the reference optimal objective $f^* = \min_{x\in X}\sum_{i=1}^m f_i(x)$ is computed using centralized Nesterov's accelerated gradient method [49, 50]. In the two plots, we observe that both $f(z(T)) - f^*$ and disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ decays to 0 as justified by our theoretical analysis in Section 1.3. In general, we observe that delays

with larger bound B and/or larger standard deviation σ in stochastic gradient yield slower convergence, as expected.

We also tested on two different objective functions: robust least squares and logistic regression. In robust least squares, we apply (1.2) to the decentralized optimization problem (1.1) where the objective function is set to

$$f_i(x) := \sum_{j=1}^{p_i} h_i^j(x), \text{ where } h_i^j(x) = \begin{cases} \frac{1}{2}|(a_i^j)^\top x - b_i^j|^2 & \text{if } |(a_i^j)^\top x - b_i^j| \le \delta \\ \delta(|(a_i^j)^\top x - b_i^j| - \frac{\delta}{2}) & \text{if } |(a_i^j)^\top x - b_i^j| > \delta \end{cases} \tag{1.43}$$

where $(a_i^j)^\top \in \mathbb{R}^n$ is the j-th row of matrix $A_i \in \mathbb{R}^{p_i \times n}$, and $b_i^j \in \mathbb{R}$ is the j-th component of $b_i \in \mathbb{R}^{p_i}$ at each node i. In this test, we simulate network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and set A_i, b_i, m, n, R, $x_i(0)$ the same way as in the decentralized least squares test above, and set the parameter of the Huber norm in the robust least squares $\delta = 0.05$. The stochastic gradient is given by $\nabla F_i(x; \xi_i(t)) = \sum_{j=1}^{p_i} \nabla h_i^j(x) + \xi_i(t)$ where $\xi_i(t)$ is generated as before with σ set to 0.01 and 0.05. Lipschitz constants L_i and L are determined as in the previous test. The settings of η and $\tau_i(t)$ remain the same as well. The objective function $f(z(T)) - f^*$ and disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ are plotted in the middle row of Figure 1.1. In these two plots, we observe similar convergence behavior as in the test on the decentralized least squares problem above. For the decentralized logistic regression, we generate $\hat{x}$, ϵ_i and A_i the same way as before, and set $b_i = \text{sign}(A_i\hat{x} + \epsilon_i) \in \{\pm 1\}^{p_i}$ ($\text{sign}(0) := 1$). Now the objective function f_i at node i is set to

$$f_i(x) = \sum_{j=1}^{p_i} \left(\log[1 + \exp((a_i^j)^\top x)] - b_i^j (a_i^j)^\top x \right), \tag{1.44}$$

where $(a_i^j)^\top \in \mathbb{R}^n$ is the j-th row of matrix $A_i \in \mathbb{R}^{p_i \times n}$, and $b_i^j \in \mathbb{R}$ is the j-th component of $b_i \in \mathbb{R}^{p_i}$. Then we perform (1.2) to solve this problem in the network $\mathcal{G}$ above. Since $\nabla^2 f_i(x) = \sum_j [\exp((a_i^j)^\top x)/(1+\exp((a_i^j)^\top x))^2] \cdot a_i^j (a_i^j)^\top \le (1/4) \cdot \sum_j a_i^j (a_i^j)^\top = (1/4) \cdot A_i^\top A_i$, there is $\|\nabla f_i(x) - \nabla f_i(x')\| \le (1/4) \cdot \|A_i^\top A_i\| \|x - x'\|$ for all $x, x' \in \mathbb{R}^n$. Therefore we set

$L_i = \|A_i^\top A_i\|_2/4$. The settings of the delay $\tau_i(t)$, η, and initial value $x_i(0)$ remain the same as before. The stochastic error level σ is set to 0.1 and 0.5. The objective function $f(z(T)) - f^*$ and disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ are plotted in the bottom row of Figure 1.1, where similar convergence behavior as in the previous tests can be observed.

We also compared the performance of decentralized gradient descent method with and without delay and stochasticity in the gradients. In this test, we synthesized networks and data in the same way as in the decentralized least squares test above. In addition, we plotted the result of $\tau_i(t) = 0$ for all $i = 1, \ldots, m$ and $\sigma = 0$ is for comparison. These results are shown in the top row of Figure 1.2, The objective function value (top left) and disagreement (top right) both decay sightly faster when there are no delay and stochastic error as shown in Figure 1.2, which is within expectations. We further tested the performance when the network size varies. In this experiment, we used four 2D lattice networks, with sizes $m = 5^2, 10^2, 15^2, 20^2$. The size of x and A_i at each node are the same as before. The objective function value (middle left) and disagreement (middle right) both decays, while it appears that network with smaller size decays faster, as shown in Figure 1.2. To demonstrate effectiveness of asynchronous consensus, we applied EXTRA [22], a state-of-the-arts synchronous decentralized consensus optimization method, to the same data generated in decentralized least squares problem with network size $m = 100$ and $\sigma = 0$ (no stochastic error in gradients). We draw computing times of these 100 nodes as independent random variables between $[.001, .500]$ms every gradient evaluation. The synchronous algorithm EXTRA needs to wait for the slowest node to finish computation and then start a new iteration, whereas in the asynchronous algorithm (1.2) the nodes communicate with neighbors every 0.01ms using updates obtained by delayed gradients . We plotted the objective function $f(z(T)) - f^*$ and disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ versus running time in the bottom row of Figure 1.2, which show that the asynchronous updates can be more time efficient by not waiting for slowest node in each iteration.

1.4.2 Test on real data

We apply algorithm (1.2) to seismic tomography where the data is collected and then processed by the nodes (sensors) in a wireless sensor network. In brief, underground seismic activities (such as earthquakes) generate acoustic waves (we use P-wave here) which travel through the materials and are detected by the sensors placed on the ground. An explanatory picture of seismic tomography using a sensor network is shown in Figure 1.3. After data preprocessing, sensor i obtains a matrix $A_i \in \mathbb{R}^{p_i \times n}$ and a vector $b_i \in \mathbb{R}^{p_i}$, and hence an objective $f_i(x) = (1/2)\|A_i x - b_i\|^2$ for $i = 1, \ldots, m$. Here $(A_i)_{kl}$, the (k,l)-th entry of matrix A_i, is the distance that the wave generated by k-th seismic activity travels through pixel l, for $k = 1, \ldots, p_i$ (p_i is the total number of seismic activities) and $l = 1, \ldots, n$ (n is the total number of pixels in the image), and $(b_i)_k$, the k-th component of b_i, is the total time that the wave travels from the source of k-th seismic activity to the sensor i. Then x_l, the l-th component of $x \in \mathbb{R}^n$, represents the unknown "slowness" (reciprocal of the velocity of the traveling wave) at that location (pixel) l. The sensors then collaboratively solve for the image x that minimizes the sum of their objective functions, under the constraint that only neighbor nodes may communicate during the computation process, since wireless signal transmission can only occur within a limited geographical range. Once x is reconstructed from $\min_x f(x) = \sum_{i=1}^m f_i(x)$, the material (e.g., rock, sand, oil, or magma) at each pixel l can be identified by the value of x_l.

The first dataset consists of a simple and connected network G with $m = 32$ nodes where each node has 3 neighbors, and $A_i \in \mathbb{R}^{p_i \times n}$ and $b_i \in \mathbb{R}^{p_i}$ where the number of seismic events is $p_i = 512$ and the size of a 2D image x to be reconstructed is $n = 64^2 = 4096$. Since the matrix by stacking all A_i is still underdetermined, we employ an objective function with Tikhonov regularization as $f_i(x) = (1/2)(\|A_i x - b_i\|^2 + \mu\|x\|^2)$ at each node i where μ is set to 0.1. Note that more adaptive regularizers of x, such as ℓ_1 and total variation (TV) which result in a nonsmooth objective function, will be explored in future research. We apply algorithm (1.3) with bound B of delays set to 5, 10, and 20 and standard deviation σ of stochastic gradient to 0.5 and 0.05. We run our algorithm using step size $\alpha(t) = 1/(2L + 2\eta\sqrt{t})$ with

η that minimizes the constant of $1/\sqrt{T}$ term in Theorem 6. The objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m} \|y_i(T) - z(T)\|^2$ versus the iteration number T are plotted in the top row of Figure 1.4, where convergence of both quantities can be observed.

The second seismic dataset contains a connected network G of size $m = 50$ where each node has 3 neighbors, and matrices $A_i \in \mathbb{R}^{p_i \times n}$ and $b_i \in \mathbb{R}^{p_i}$ where $p_i = 800$ and the size of 3D image x to be reconstructed is $n = 32^3 = 32768$. We use the same objective function with Tikhonov regularization as before with $\mu = 0.01$. Other parameters are set the same as in the previous test on a 2D seismic image. The settings for B and σ remain the same. The objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m} \|y_i(T) - z(T)\|^2$ versus the iteration number T are plotted in the middle row of Figure 1.4, where similar convergence behavior can be observed.

The last seismic dataset consists of a connected network G of size $m = 10$ where the average node degree is 5, and matrices $A_i \in \mathbb{R}^{p_i \times n}$ and $b_i \in \mathbb{R}^{p_i}$ where $p_i = 1,816$ and the size of 3D image x to be reconstructed is $n = 160 \times 200 \times 24 = 768,000$. In this test, we employ objective $f_i(x) = (1/2)(\|A_i x - b_i\|^2 + \mu \|Dx\|^2)$ where $\mu = 0.1$ and D is the discrete gradient operator. Other parameters are set the same as in the previous two seismic datasets. The bound B of delay is set to 4, 8, and 16, and standard deviation of stochastic gradient σ is set to 1e-4 and 5e-4. The objective function $f(z(T))$ and disagreement $\sum_{i=1}^{m} \|y_i(T) - z(T)\|^2$ versus the iteration number T are plotted in the last row of Figure 1.4. The reconstructed image is displayed in the right panel of Figure 1.5. By comparing with the solution obtained by centralized LSQR solver (left), we can see the image is faithfully reconstructed on a decentralized network with delayed stochastic gradients.

1.5 Concluding Remarks

We analyzed the convergence of decentralized delayed stochastic gradient descent method as in (1.2) for solving the consensus optimization (1.1). The algorithm takes into consideration that the nodes in the network privately hold parts of the objective function and collaboratively solve for the consensus optimal solution of the total objective while they

can only communicate with their immediate neighbors, as well as the delays of gradient information in real-world networks where the nodes cannot be fully synchronized. We show that, as long as the random delays are bounded in expectation and a proper diminishing step size policy is employed, the iterates generated by the decentralized gradient decent method converge to a consensus solution. Convergence rates of both objective and consensus were derived. Numerical results on a number of synthetic and real data were also presented for validation.

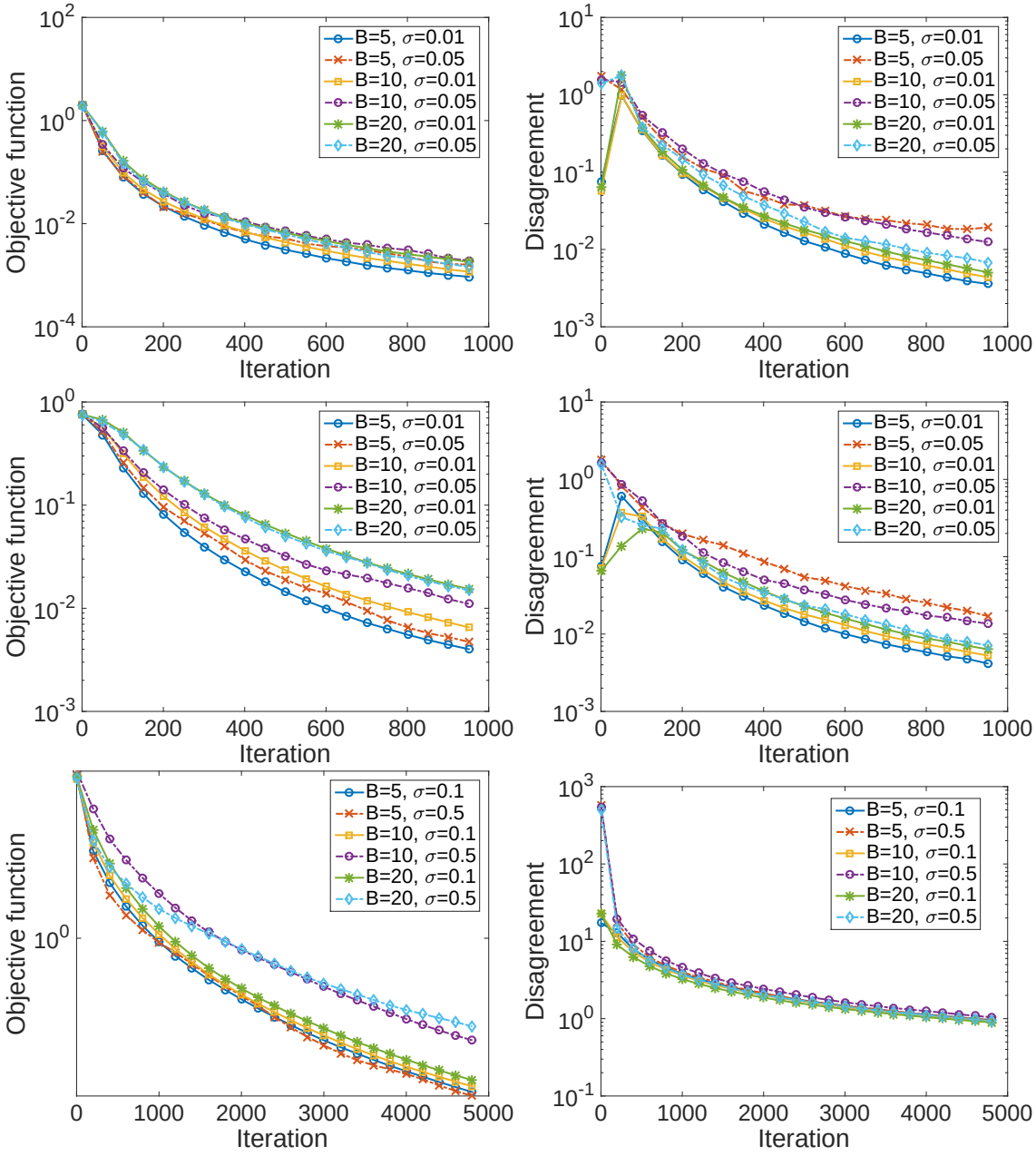

Figure (1.1) Test on synthetic decentralized least-squares (top), robust least-squares (middle), and logistic regression (bottom) for different levels of delay B and standard deviation in stochastic gradient σ. Left: objective function $f(z(T)) - f^*$ versus iteration number T, where $f^* = f(x^*)$ is the optimal value. Right: disagreement $\sum_{i=1}^m \|y_i(T) - z(T)\|^2$ versus iteration number T.

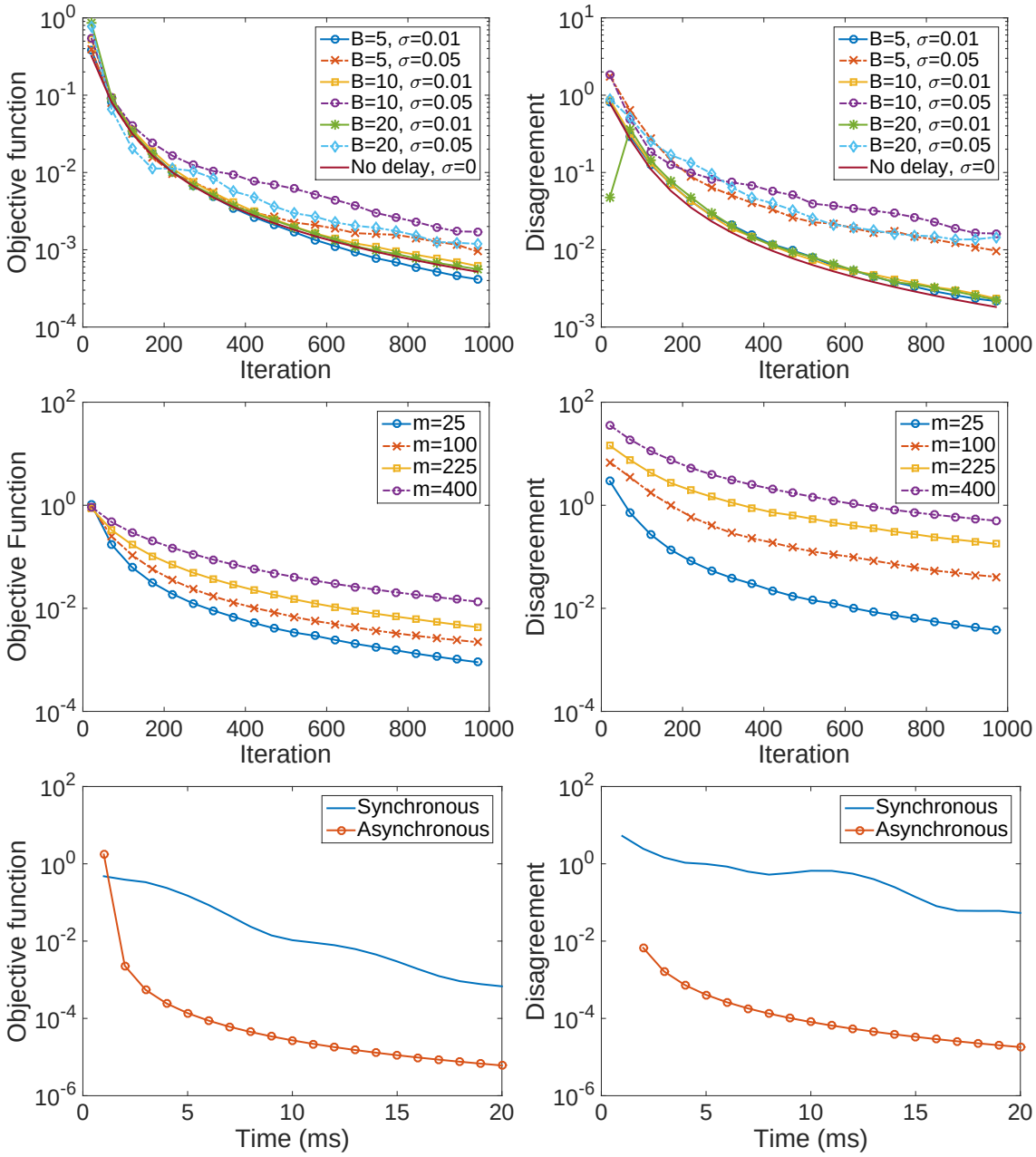

Figure (1.2) Test on synthetic decentralized least-squares with and without delay/stochasticity (top) and varying network size (bottom). Left: objective function $f(z(T))$ versus iteration number T. Right: disagreement $\sum_{i=1}^{m} \|y_i(T) - z(T)\|^2$ versus iteration number T.

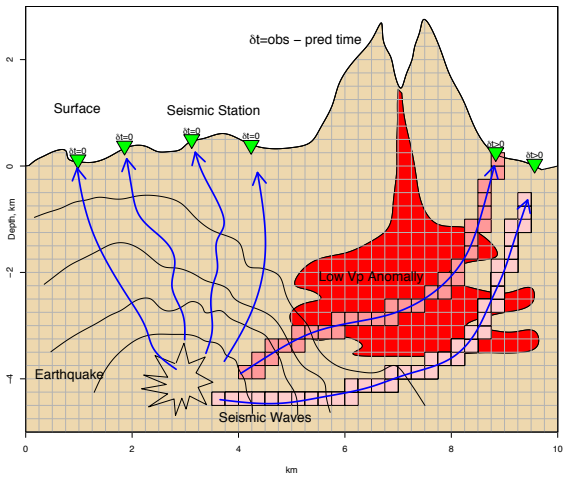

Figure (1.3) Seismic tomography of an active volcano using wireless sensor network. When there is a seismic activity (e.g., an earthquake) happens underground, its acoustic waves (blue solid curves with arrows) travel to the ground surface and are detected by the sensors (green triangles). Then the sensors communicate wirelessly to reconstruct the entire image, where each square (tan, pink or red) represents a pixel of the image $x \in \mathbb{R}^n$.

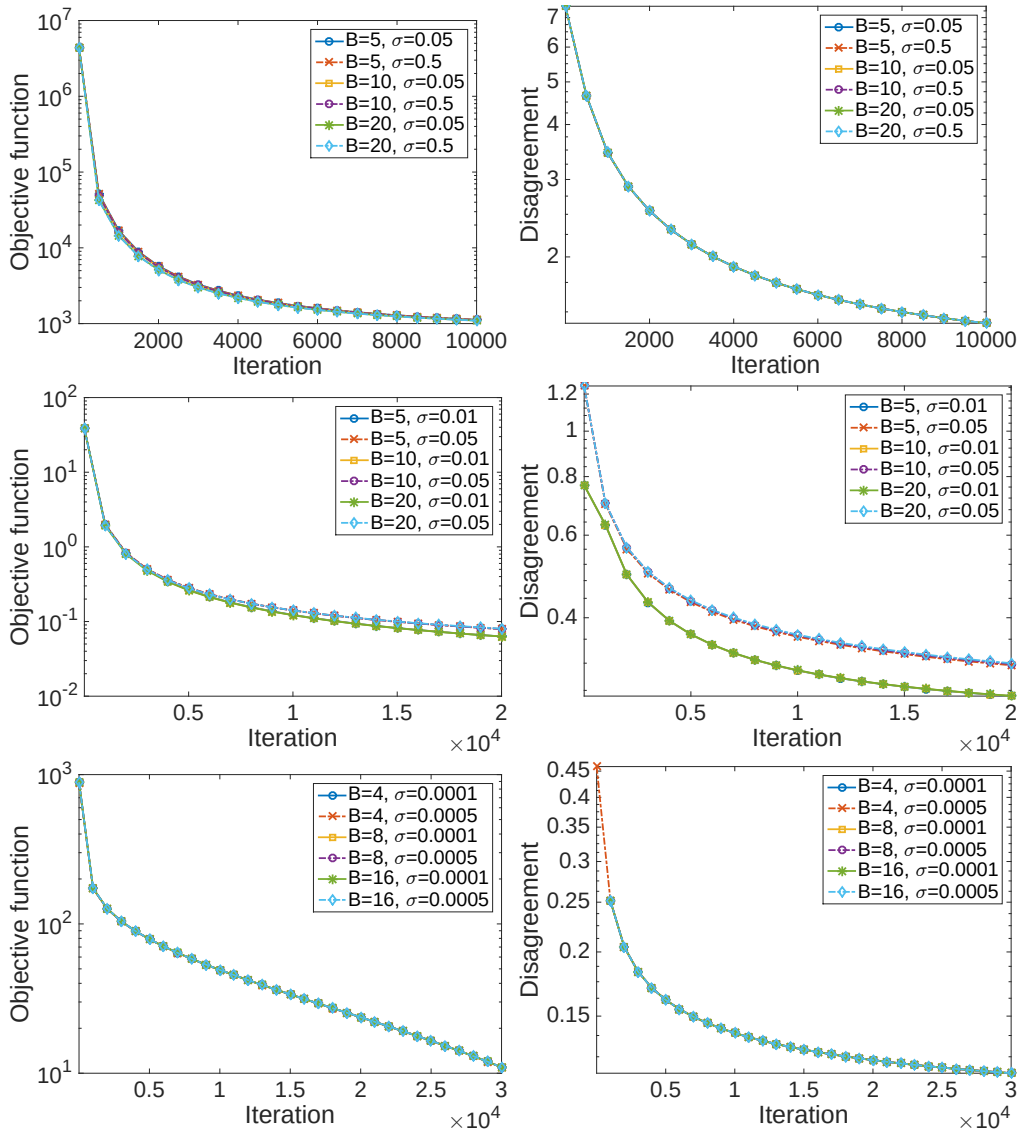

Figure (1.4) Tests on real seismic image reconstruction problems with $2D$ image with $n = 64^2$ (top), $3D$ image with $n = 32^3$ (middle), and $3D$ image with $n = 160 \times 200 \times 24$ (bottom) for different levels of delay B and standard deviation in stochastic gradient σ. Left: objective function $f(z(T))$ versus iteration number T. Optimal value indicates $f^* := f(x^*)$. Right: disagreement $\sum_{i=1}^{m} \|y_i(T) - z(T)\|^2$ versus iteration number T.

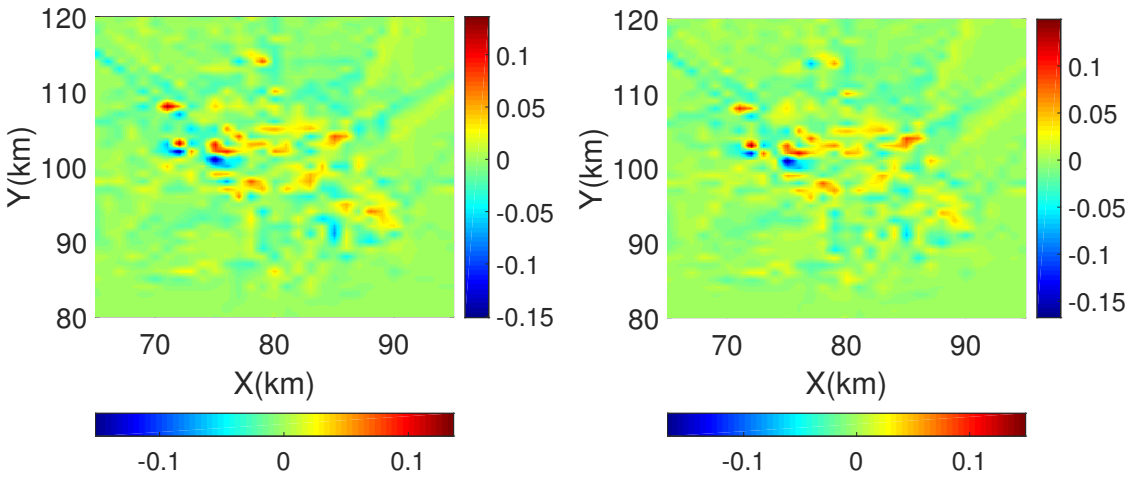

Figure (1.5) Cross section of a reconstructed 3D seismic image generated by a centralized LSQR solver (left) and decentralized algorithm with delayed stochastic gradient (1.2) with $B = 4$ and $\sigma = 10^{-4}$ (right).

PART 2

PARAMETER ESTIMATION FOR CYCLIC DISEASES

2.1 Introduction

We now consider whether decentralized consensus methods may be extended to parameter estimation problems in epidemiology. Parameter estimation problems in this field arise from the need for researchers to accurately recover and predict incidence cases for diseases. One common method for modeling disease epidemic outbreaks is to use a compartmental model [51], originating with the work in [52]. The model we consider is the SEIR model, where individuals within the population are assigned to different groups in the model: S - susceptible, E - exposed, I - infected, and R - recovered. The model is given as a system of differential equations, with members of the population entering and exiting the different categories being modeled as derivatives (with respect to time) dependent on the other categories of the model, which are themselves time-dependent. Parameters for the system are given as coefficients for the different categories and vary depending on the specific disease under consideration. Compartmental coefficients specific to some common diseases have been estimated from prior outbreaks and are taken as a priori input to the SEIR model being considered [53].

In addition to accurate selection of compartmental parameters (category coefficients in the system of ODE's), accurate estimation of the disease transmission rate is another vital part of accurately recovering and forecasting incidence cases of a given disease [54, 55, 56]. Disease incidence cases vary with time, and the underlying transmission mechanisms are believed to be influenced by weather, human contact rates, and environmental or other changes [57, 58]. When considering the incidence cases for cyclic diseases (influenza, pneumonia, etc.), time-dependent transmission rates are used to account for the seasonal spikes that are observed [56, 59]. In general, recovering the transmission rate from observed incidence case

data is an ill-posed (inverse) problem, and the selection of an appropriate numerical method is critical [60]. The authors in [61] describe one method for recovering time-dependent transmission rate given incidence case data, but do not conduct incidence case forecasting with their recovered transmission rate.

In order to recover the desired parameters numerically, iterative schemes are commonly used. One of the more commonly known iterative schemes in optimization is Newtons method, which requires the computation of the Jacobian in each iteration. As this is generally a difficult and expensive computational operation, a modified version of Newton's method was introduced by C.G. Broyden in [62], whereby the Jacobian is computed once upon initialization of the iterative scheme and is then subjected to rank-one updates in each iteration. However, the presence of noise in the observed data leads to increased sensitivity of the solution dependent on the input, and so regularization methods are required to obtain convergence. In [63], the author introduces a modification of Broydens method with the regularization being done by smoothing the input data and by stopping the iterative process at an appropriate index. In [64], the author extends the result from [63] by removing a nonlinearity assumption, leading to an algorithm that is applicable to a broader class of nonlinear inverse problems.

2.1.1 Model

We consider the standard SEIR compartmental model of epidemiology in our approach. In order to account for the cyclic nature of the diseases we are interested in, we suppose that disease transmission rate is time dependent, so that $\beta = \beta(t)$.

We consider the following system of ODE's:

$$\frac{dS}{dt} = \mu N - \beta(t)S(t)\frac{I^{\alpha}(t)}{N} - \mu S(t) + \sigma R(t) \tag{2.1}$$

$$\frac{dE}{dt} = \beta(t)S(t)\frac{I^{\alpha}(t)}{N} - \mu E(t) - \kappa E(t) \tag{2.2}$$

$$\frac{dI}{dt} = \kappa E(t) - \gamma I(t) - \mu I(t) \tag{2.3}$$

$$\frac{dR}{dt} = \gamma I(t) - \mu R(t) - \sigma R(t) \tag{2.4}$$

with initial conditions

$$S(0) = S_0, \quad E(0) = E_0, \quad I(0) = I_0, \quad R(0) = R_0 \tag{2.5}$$

and parameters

Table (2.1) System parameters

Parameter	Definition
N	Total population
$1/\kappa$	Average incubation period
$1/\gamma$	Average time from symptom onset to recovery
$1/\sigma$	Average time for the loss of immunity
μ	Birth/death rate
α	Scaling component

2.1.2 Method

We wish to forecast future disease incidence cases given past incidence case data, and we do so by considering the constrained least-squares minimization problem

$$\min_{\beta,S,E,I,R} \ \frac{1}{2}\,||\kappa E[\beta] - D||^2 \quad \text{subject to} \quad F(\beta, S, E, I, R) = 0, \tag{2.6}$$

where $D = [D_1, D_2, ..., D_m]^\top$ is a vector of (daily, weekly, monthly) incidence cases. We aim to recover $\beta(t)$ given D, and use this $\beta(t)$ in system (2.1) - (2.4) to predict future incidence cases. In order to implement this numerically, we must approximate the transmission rate $\beta(t)$ as a linear combination of known basis elements, which we refer to as discretization. To ensure we do not bias our results towards one particular choice of basis, we consider the effects of discretizing $\beta(t)$ using two different bases.

Discretizing transmission rate with a trigonometric basis. One way of recovering $\beta(t)$ numerically is to discretize it by projecting it onto the subspace spanned by a basis containing cyclic functions, having the form $\{s_n(u(t)), c_n(u(t)) : 1 \leq n \leq \overline{N}\} \cup \{1\}$ for some $\overline{N} \in \mathbb{N}$, where $s_n(u(t)) = 0.15\left[\sin(2\pi n \cdot u(t)/L) + 1.5\right]$, and $c_n(u(t)) = 0.15\left[\cos(2\pi n \cdot u(t)/L) + 1.5\right]$ for all n and some $L \in \mathbb{N}$. To maintain consistency in our basis across a variety of epidemic lengths, we adopt the following convention for $s_n(u(t))$ and $c_n(u(t))$. Suppose an epidemic starts at $t = a$, and ends at $t = b$. We linearly interpolate the interval (a, b) onto the interval $(0, 1)$ by defining $u(t) = (t - a)/(b - a)$. This allows us to control the number of periods appearing in (a, b) for any epidemic length, simply by our choice of L and by our choice of basis size. To elaborate slightly, for $L = 1$, $s_1(u(t))$ and $c_1(u(t))$ will each complete one period as t ranges from a to b; for $L = 2$, $s_1(u(t))$ and $c_1(u(t))$ will each complete a half period as t ranges from a to b; and so on. With $u(t)$ so defined, and for t ranging from a to b, we have the following approximation for transmission rate $\beta(t)$:

$$\beta(t) \approx \hat{\beta}(A, t) := a_0 + \sum_{n=1}^{\overline{N}} a_{2n-1} s_n(u(t)) + a_{2n} c_n(u(t))$$

where we define $A = [a_0, a_1, a_2, \cdots, a_{2\overline{N}}] \in \mathbb{R}^{2\overline{N}+1}$.

Discretizing transmission rate with a basis of Legendre polynomials. The next basis under consideration is the basis of Legendre polynomials $P_0(u(t)), P_1(u(t)), \cdots, P_{\overline{N}}$ for some $\overline{N} \in \mathbb{N}$, where $P_n(u(t))$ is the Legendre polynomial of degree n. As with the trigonometric basis from Section 2.1.2, we desire to keep our function behavior consistent across a

variety of epidemic lengths. As such, we linearly interpolate the epidemic interval similarly to how the epidemic interval was interpolated for the selected trigonometric basis. In the case of Legendre polynomials, we choose to interpolate the epidemic interval from $t \in [a, b]$ to $u(t) \in [-1, 1]$. Thus, for any week $t \in [a, b]$ we arrive at the input $u(t) = 2(t-a)/(b-a) - 1$ for any Legendre polynomial P_n. With $u(t)$ defined in this manner, we arrive at the following approximation for the transmission rate $\beta(t)$:

$$\beta(t) \approx \hat{\beta}(A, t) := \sum_{n=0}^{\overline{N}} a_n P_n(u(t))$$

where we define $A = [a_0, \cdots, a_{\overline{N}}]$.

Numerical approach. Now we present our numerical approach, which is independent of our choice of basis. Suppose $\big[S(A,t), E(A,t), I(A,t), R(A,t)\big]$ is a numerical solution to the system

$$\frac{dS}{dt} = \mu N - \hat{\beta}(A,t)S(t)\frac{I^{\alpha}(t)}{N} - \mu S(t) + \sigma R(t) \tag{2.7}$$

$$\frac{dE}{dt} = \hat{\beta}(A,t)S(t)\frac{I^{\alpha}(t)}{N} - \mu E(t) - \kappa E(t) \tag{2.8}$$

$$\frac{dI}{dt} = \kappa E(t) - \gamma I(t) - \mu I(t) \tag{2.9}$$

$$\frac{dR}{dt} = \gamma I(t) - \mu R(t) - \sigma R(t) \tag{2.10}$$

$$S(0) = S_0, \quad E(0) = E_0, \quad I(0) = I_0, \quad R(0) = R_0. \tag{2.11}$$

Given incidence case data D, we turn to the unconstrained least squares minimization problem

$$\min_A \frac{1}{2} ||\Phi(A) - D||^2 \tag{2.12}$$

where $\Phi(A) = \kappa E(A)$.

To recover the vector of expansion coefficients A, we proceed to solve (2.12) iteratively, using the following regularized version of Broyden's secant method for solving nonlinear operator equations:

$$\begin{aligned} A^{k+1} &= A^k + \psi\big[(I - P^k)(A^{k-1} - A^k) - Q^k F(A^k)\big] \\ J^{k+1} &= J^k + \frac{\langle s^k, \cdot \rangle}{\|s^k\|^2}\left(y^k - J^k s^k\right) \end{aligned} \tag{2.13}$$

where $\psi > 0$, $s^k = A^{k+1} - A^k$, and $y^k = F(A^{k+1}) - F(A^k)$. Here Q^k is a regularized pseudo-inverse of J^k, J^0 is an initial approximation to the jacobian of operator $\Phi(A)$, and

$$P^k = \sum_{j=1}^{\lambda^k} \langle \cdot, u_j^k \rangle u_j^k$$

is the orthogonal projector onto the subspace spanned by the first λ^k eigenvectors of $(J^k)^* J^k$. More specifically, Q^k is constructed by filtering out singular values below a given threshold $\sqrt{\epsilon}$ for some $\epsilon > 0$:

$$Q^k = \sum_{j \in \Lambda} \frac{\sigma(\epsilon^k, \mu_j^k)}{\mu_j^k} \langle \cdot, v_j^k \rangle u_j^k, \text{ where } \sigma(\epsilon, \mu) = \begin{cases} 1 & \mu \geq \sqrt{\epsilon} \\ 0 & \mu < \sqrt{\epsilon} \end{cases} \tag{2.14}$$

Once a vector of expansion coefficients is recovered, we can estimate $\beta(t) \approx \hat{\beta}(A, t)$ and use this approximation in (2.7) - (2.10) to produce an estimate for future incidence cases.

2.2 Testing the method on simulated data

2.2.1 Generating synthetic data

To test our method, we generate synthetic incidence case data for an outbreak on a simulated population. Doing so will require solving system (2.1) - (2.4) and using the resulting E vector as the incidence case data to be used. To solve this system, we define a model transmission rate $\beta(t)$ that exhibits oscillating behavior. Specifically, we let

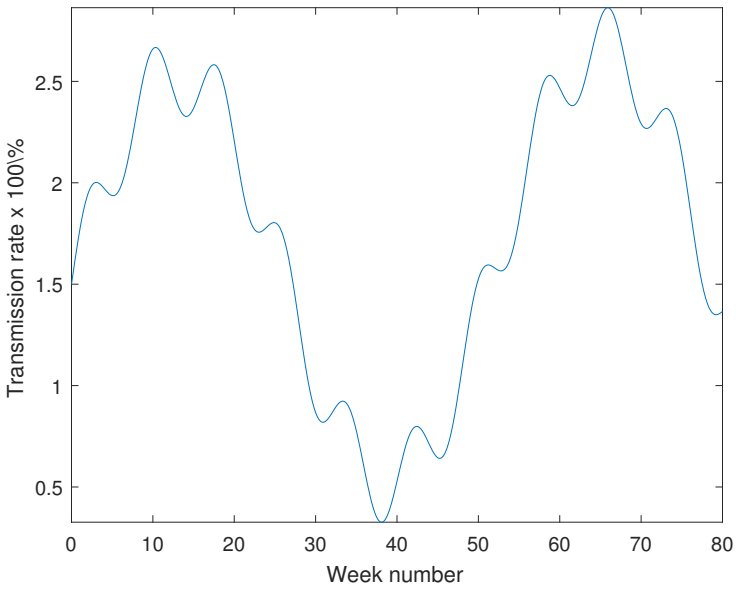

Figure (2.1) Plot of model transmission rate.

$\beta(t) = \big(\sin(\pi t/26) + 1.5\big) \cdot \exp(t/1000) + .2\sin(\pi t/4)$. The choice of these specific factors and constants is not of particular importance (and neither is this specific format of transmission rate function), except to ensure that the selected model transmission rate reasonably mimics a generally cyclic nature observed in past real-life disease outbreaks with periodic spikes in incidence cases. The selected model transmission rate is shown in Figure 2.1.

We set the initial conditions $S_0 - R_0$ in system (2.1) - (2.4) as follows

$$S(0) = 9975, \quad E(0) = 0, \quad I(0) = 25, \quad R(0) = 0$$

and we fix the system parameters as defined in Table 2.2.

Table (2.2) Simulation system parameters

Parameter	Value	Definition
N	10,000	Total population
$1/\kappa$	8/7	Average incubation period
$1/\gamma$	5/7	Average time from symptom onset to recovery
$1/\sigma$	150/7	Average time for the loss of immunity
μ	.025	Birth/death rate
α	0.85	Scaling component

With the selected transmission rate, system parameters, and initial conditions, we set the simulated outbreak length to $m = 80(= b-a)$ and we run MATLAB's ode23s ODE solver on system (2.1) - (2.4) to simulate incidence case data, scaling the resulting E values by κ. As the resulting incidence cases show smooth, non-erratic changes from week to week, we add noise to simulate the results obtained by real-world data collection efforts. We generate noise according to the poisson distribution, with poisson parameter at each week of the simulated epidemic proportional to the incidence cases observed at that point. For instance, if the number of simulated incidence cases in week 10 is 150 cases, the poisson parameter is $c \cdot 150$, where $c \in (0, 1)$. We find that scaling by a value smaller than 1 produces data that resembles the type observed when comparing to real-life data collection efforts. We randomly select a value from the poisson distrubution with this given parameter, and then add or subtract this value (randomly add or subtract with equal probability) from each incidence case datapoint. This generates data which is taken to be our noisy, "real-world" incidence case data. The comparison between "clean" data and "ground truth" data is shown in Figure 2.2. This noisy incidence case data vector is taken to be the data vector D for this simulation.

2.2.2 Recovering the transmission rate

Once we have an incidence case data vector to work with, we test our ability to forecast the given incidence cases by using only part of the incidence case data vector. We do this to mimick the real-world scenario in which one would like to produce a forecast of future

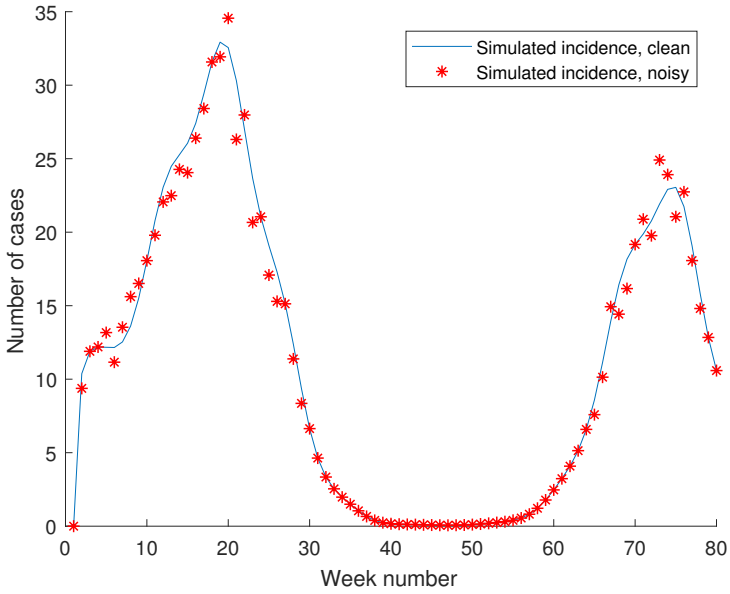

Figure (2.2) Clean vs. noisy simulated incidence cases.

incidence cases given incidence cases observed up to this point. Comparing our forecast to our synthetic results will give us an idea of the accuracy and power of our forecast.

To generate a forecast, we first need to estimate the transmission rate, which is estimated by solving problem (2.12) iteratively for A, the vector of expansion coefficients for the discretized transmission rate estimate. Once we have recovered A, we can estimate the transmission rate $\hat{\beta}(A,t)$ and we use this estimate in system (2.7) - (2.10). We then use MATLAB's built-in ode23s solver to solve the system, producing an incidence case data vector which will serve as our forecasted values.

For this simulation, we conduct 5 separate forecasts, using 42, 44, 46, 48, and 50 weeks of data respectively. For the first forecast, we take the first 42 data points of D as our measured data, i.e. we suppose these are the reported cases of a disease epidemic occuring within the most recent 42 weeks. We take the first 44 points for the 2nd forecast, 46 for the 3rd, and so on.

To initialize the iterative process for each forecast, we start with an initial guess for the coefficients of $\hat{\beta}(A,t)$, starting at A_0. To not bias our results to this particular transmission

rate, we randomly select a constant λ from the uniform distribution on $[\Lambda_1, \Lambda_2]$ for some $\Lambda_1, \Lambda_2 \in \mathbb{N}$, where $0 \leq \Lambda_1 \leq \Lambda_2 - 1$, to serve as the initial (constant) guess for transmission rate $\hat{\beta}_0 = \hat{\beta}(A_0, t)$. Then the expansion coefficients A_0 are chosen so that $\hat{\beta}_0 \approx \lambda$ for all $t \in [a, b]$. As an additional check against bias, we repeat this sampling process multiple times for each partial data length. That is, for any given partial data length in our current simulation, we sample $\lambda \in [\Lambda_1, \Lambda_2]$ 20 times, producing an incidence case forecast for each estimated transmission rate $\hat{\beta}_K$ resulting from each choice of λ. For each partial data length, we obtain 20 estimates of β and 20 incidence case forecasts. We then take the mean of both estimated transmission rates and incidence case forecasts. We plot all $\hat{\beta}_K$ estimates on the same axis, and compare with the plot of our model transmission rate β to compare accuracy of recovery. We do the same for incidence case forecasts.

Using a basis of Trigonometric functions. We set the number of basis pairs to $\overline{N} = 10$, yielding a coefficient vector of length $2\overline{N} + 1 = 21$. We set the maximum number of iterations for our algorithm (regularized Broyden's method) to $K = 60$, we set the step size $\psi = 0.1$, and the singular value truncation threshold for Q^k to $\sqrt{\epsilon} = 3$. We find that a higher truncation threshold helps balance accuracy with the threat of overfitting to noisy data. After running the algorithm for the specified number of iterations K, we use the arrived at coefficient vector A_K to estimate $\beta(t) \approx \hat{\beta}(A_K, t) := \hat{\beta}_K$ and use this estimate in system $(2.7) - (2.10)$. Solving the system produces an incidence case data vector which is taken to be our estimate produced by this model. As we are using partial data from D for each forecast, we expect the estimated incidence case data to closely follow the observed incidence up to the current point (matching the observed incidence cases), and we view the incidence cases estimated for future dates (dates beyond the length of our partial data vector D) as the forecasted incidence cases produced by our chosen model and estimated transmission rate $\hat{\beta}_K$. The results are plotted in Figures 2.3 - 2.8.

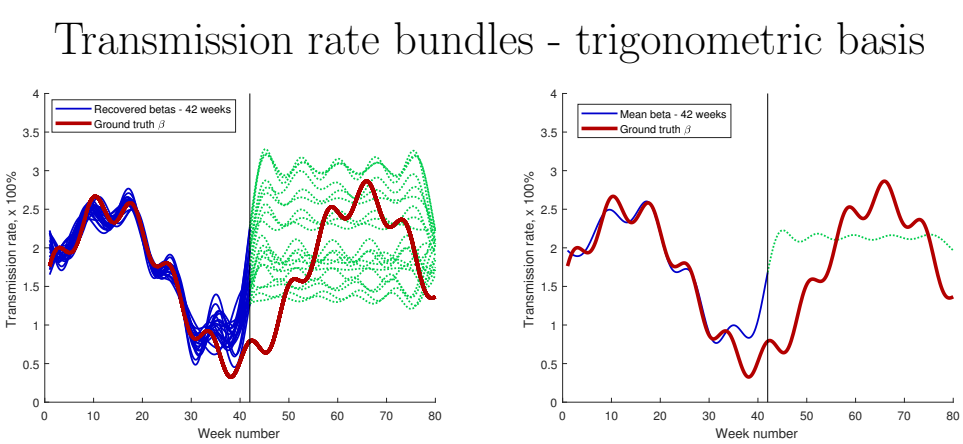

Figure (2.3) Transmission rates estimated using 42 weeks of data.

Left: Each $\hat{\beta}_K$ is plotted vs. the "ground truth" model transmission rate β. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the forecasted transmission rate (the rate where data was not available). The vertical line delineates where the transmission rate estimate is no longer based on observed data. Right: The mean of all $\hat{\beta}_K$ is plotted vs. the model transmission rate β.

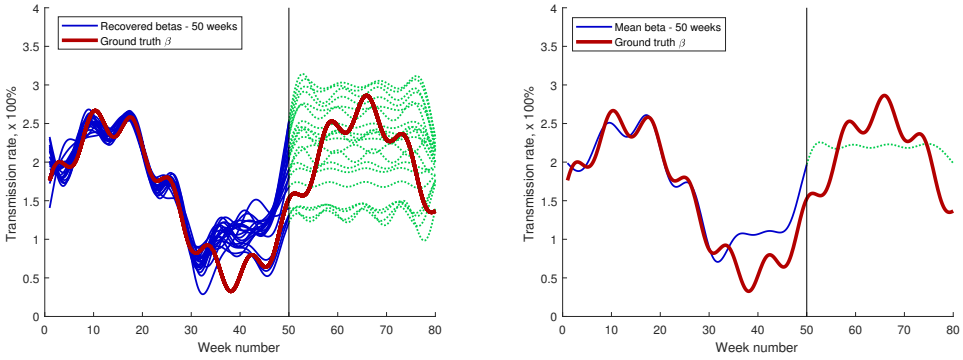

Figure (2.4) Transmission rates estimated using 50 weeks of data.

Left: Each $\hat{\beta}_K$ is plotted vs. the "ground truth" model transmission rate β. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the forecasted transmission rate (the rate where data was not available). The vertical line delineates where the transmission rate estimate is no longer based on observed data. Right: The mean of all $\hat{\beta}_K$ is plotted vs. the model transmission rate β.

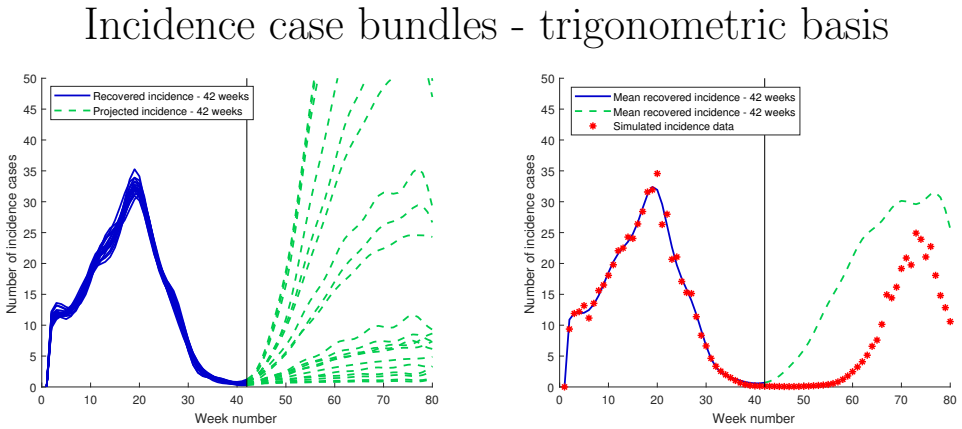

Figure (2.5) Incidence cases recovered using 42 weeks of data.

Left: Incidence case predictions produced by distinct choices of initial choice of $\hat{\beta}_0$. The solid lines indicate values where partial data was used for recovery, while the dashed lines show the forecasted incidence cases. The vertical line delineates where the incidence case data transitions from being fitted to observed data to being a forecast of predicted incidence cases. Right: The mean of all resultant incidence cases for the given partial data length vs. the simulated incidence case data used for the simulation.

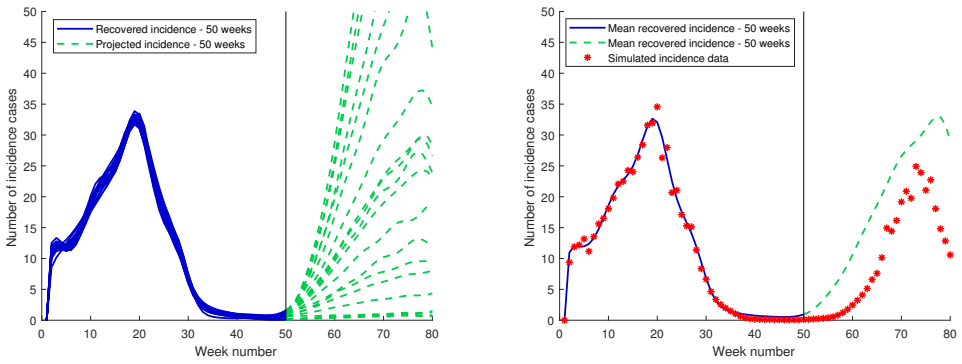

Figure (2.6) Incidence cases recovered using 50 weeks of data.

Left: Incidence case predictions produced by distinct choices of initial choice of $\hat{\beta}_0$. The solid lines indicate values where partial data was used for recovery, while the dashed lines show the forecasted incidence cases. The vertical line delineates where the incidence case data transitions from being fitted to observed data to being a forecast of predicted incidence cases. Right: The mean of all resultant incidence cases for the given partial data length vs. the simulated incidence case data used for the simulation.

Incidence case forecasts - trigonometric basis

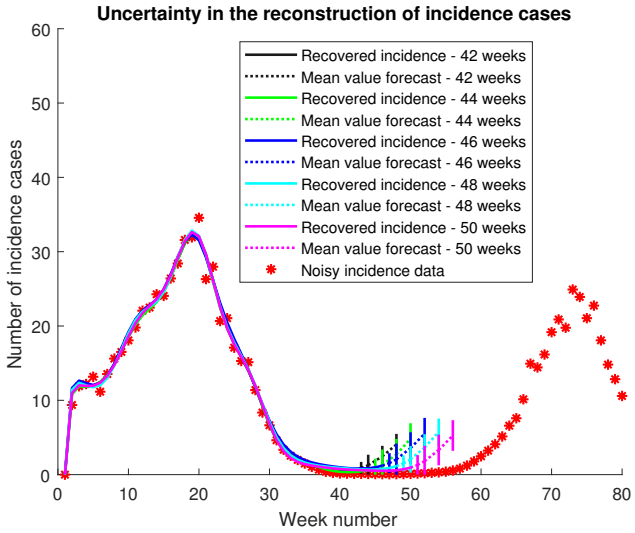

Figure (2.7) Incidence case projections with 95 percent confidence intervals.

Projections are made in 2 week increments past observed incidence case data, with vertical bars showing the upper and lower bounds of the 95% confidence interval for incidence case forecasts.

Incidence case forecasts - trigonometric basis

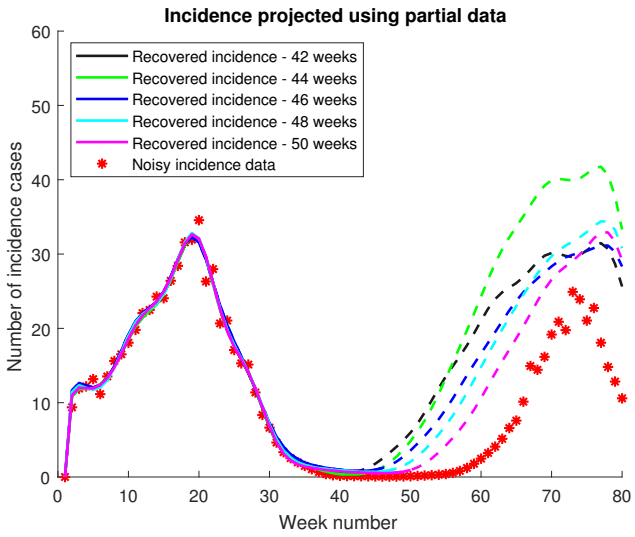

Figure (2.8) Incidence case forecasts.

The mean of all resultant incidence cases for all partial data lengths compared. We notice that the recovered incidence cases predicted by our model align very closely with the observed incidence case data up to the point where data was available. Forecasts are seen to deviate more where observed data is not available.

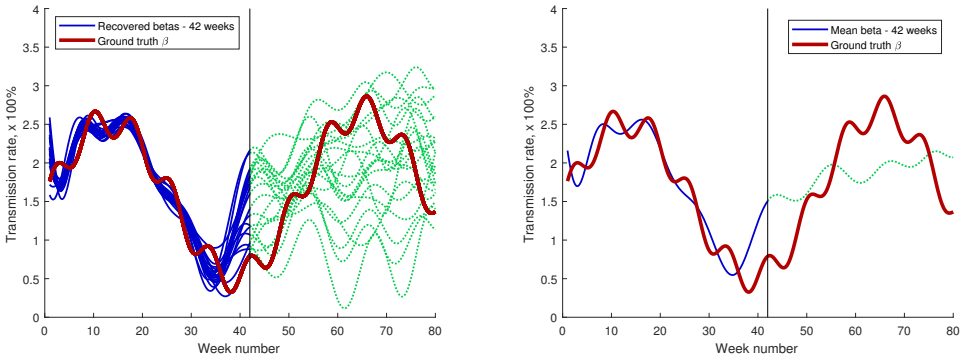

Figure (2.9) Transmission rates estimated using 42 weeks of data.

Left: Each $\hat{\beta}_K$ is plotted vs. the "ground truth" model transmission rate β. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the forecasted transmission rate (the rate where data was not available). The vertical line delineates where the transmission rate estimate is no longer based on observed data. Right: The mean of all $\hat{\beta}_K$ is plotted vs. the model transmission rate β.

Using a basis of Legendre polynomials. We set the number of Legendre polynomials in the discretization basis to $\overline{N} = 30$. We set the maximum number of iterations for our algorithm (regularized Broyden's method) to $K = 120$, we set the step size $\psi = 0.1$, and the singular value truncation threshold for Q^k to $\sqrt{\epsilon} = 3$. We find that a higher truncation threshold helps balance accuracy with the threat of overfitting to noisy data. After running the algorithm for the specified number of iterations K, we use the arrived at coefficient vector A_K to estimate $\beta(t) \approx \hat{\beta}(A_K, t) := \hat{\beta}_K$ and use this estimate in system (2.7) – (2.10). Solving the system produces an incidence case data vector which is taken to be our estimate produced by this model. As we are using partial data from D for each forecast, we expect the estimated incidence case data to closely follow the observed incidence up to the current point (matching the observed incidence cases), and we view the incidence cases estimated for future dates (dates beyond the length of our partial data vector D) as the forecasted incidence cases produced by our chosen model and estimated transmission rate $\hat{\beta}_K$. The results are plotted in Figures 2.9 - 2.14.

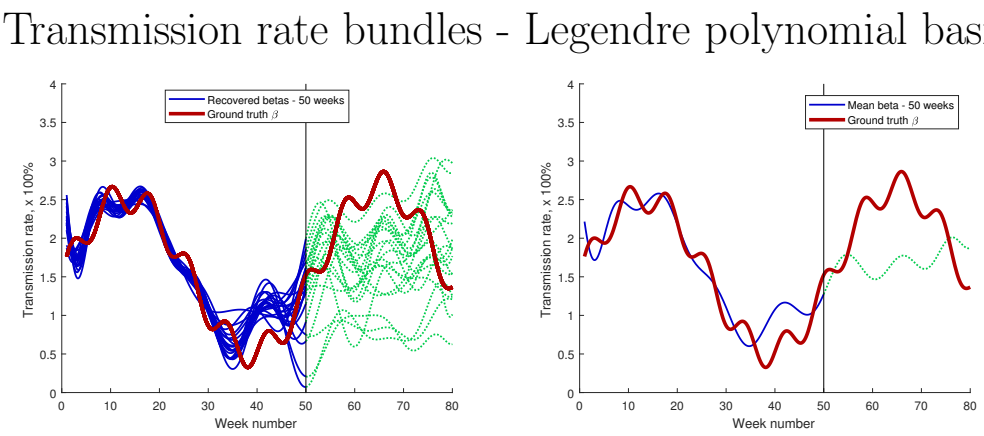

Figure (2.10) Transmission rates estimated using 50 weeks of data.

Left: Each $\hat{\beta}_K$ is plotted vs. the "ground truth" model transmission rate β. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the forecasted transmission rate (the rate where data was not available). The vertical line delineates where the transmission rate estimate is no longer based on observed data. Right: The mean of all $\hat{\beta}_K$ is plotted vs. the model transmission rate β.

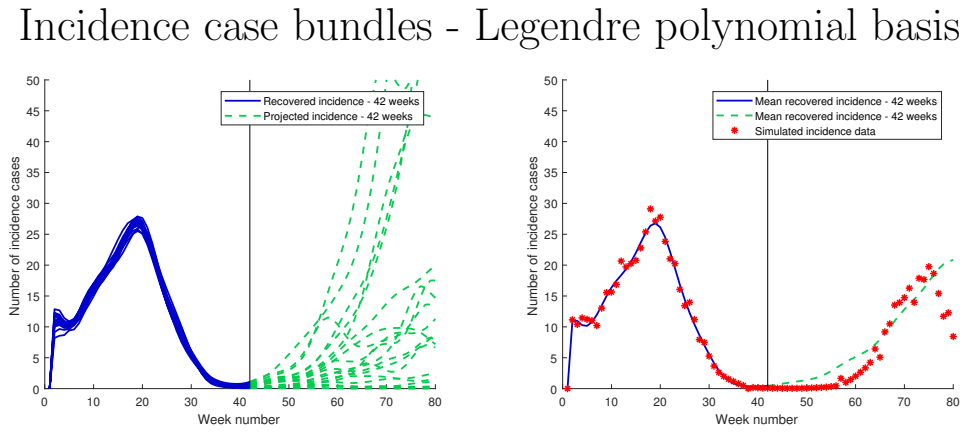

Figure (2.11) Incidence cases recovered using 42 weeks of data.

Left: Incidence case predictions produced by distinct choices of initial choice of $\hat{\beta}_0$. The solid lines indicate values where partial data was used for recovery, while the dashed lines show the forecasted incidence cases. The vertical line delineates where the incidence case data transitions from being fitted to observed data to being a forecast of predicted incidence cases. Right: The mean of all resultant incidence cases for the given partial data length vs. the simulated incidence case data used for the simulation.

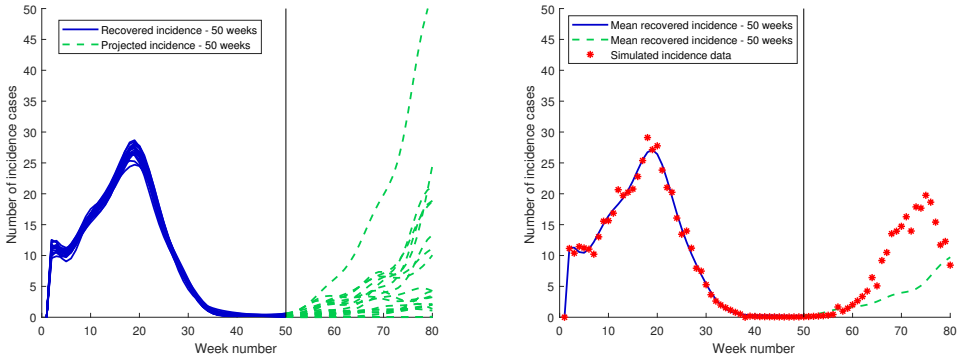

Figure (2.12) Incidence cases recovered using 50 weeks of data.

Left: Incidence case predictions produced by distinct choices of initial choice of $\hat{\beta}_0$. The solid lines indicate values where partial data was used for recovery, while the dashed lines show the forecasted incidence cases. The vertical line delineates where the incidence case data transitions from being fitted to observed data to being a forecast of predicted incidence cases. Right: The mean of all resultant incidence cases for the given partial data length vs. the simulated incidence case data used for the simulation.

2.3 Testing the method on real data

2.3.1 Data source

We now test our method on real data collected from a previous measles outbreak that occured in the UK. We consider outbreaks in 3 different cities to not bias our results towards one specific outbreak, considering the outbreaks in Birmingham, Newcastle, and London. All data for this test was collected either from OPCS (Office of Population Censuses and Surveys) reports, from the Registrar General's Quarterly or Annual Reports, or from various English census reports, as referenced in [65]. The cases under consideration were recorded weekly from 1948 through 1950, covering 3 years of observational data starting with January 17th, 1948. This particular interval of time was selected so as to include a window where pronounced spikes in incidence cases are observed. For our purposes, we desired to include a large enough window so that multiple spikes are seen, while keeping the window as small as possible so as to justify a constant population size.

Incidence case forecasts - Legendre polynomial basis

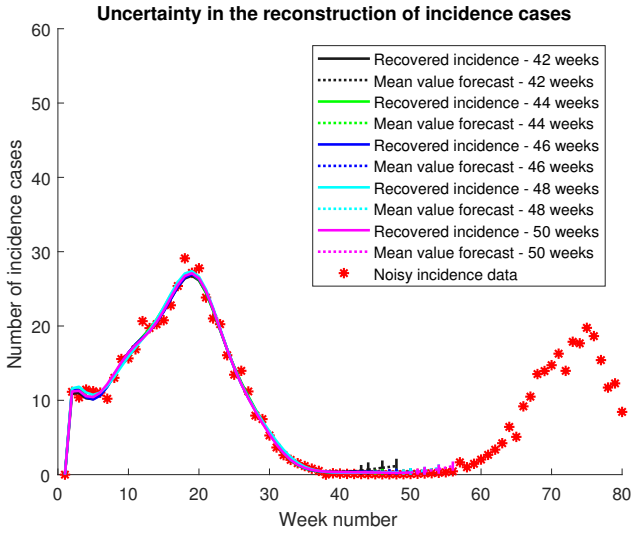

Figure (2.13) Incidence case projections with 95 percent confidence intervals.

Projections are made in 2 week increments past observed incidence case data, with vertical bars showing the upper and lower bounds of the 95% confidence interval for incidence case forecasts.

Incidence case forecasts - Legendre polynomial basis

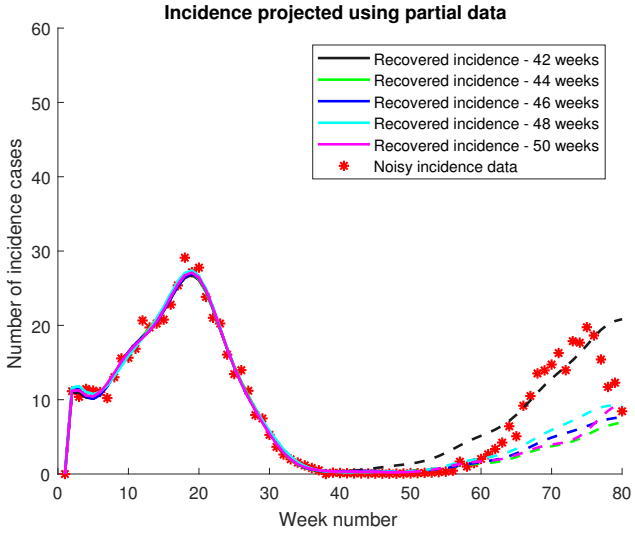

Figure (2.14) Incidence case forecasts.

The mean of all resultant incidence cases for all partial data lengths compared. We notice that the recovered incidence cases predicted by our model align very closely with the observed incidence case data up to the point where data was available. Forecasts are seen to deviate more where observed data is not available.

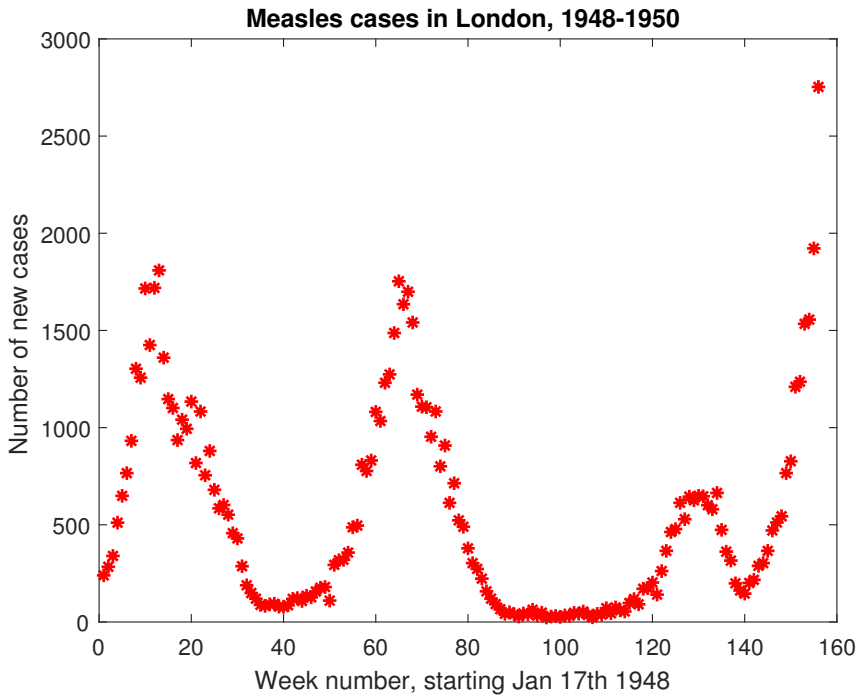

Figure (2.15) Weekly measles incidence cases, London measles outbreak 1948-1950

A plot of recorded incidence cases in London is shown in Figure 2.15. For each city, we use the best known estimate for the population at the time, as determined by the British census circa 1950.

2.3.2 Estimating the transmission rate

With this particular dataset and observation window, we use the first year of data as a given partial data vector to see how well our method's predicted incidence cases line up with the recorded number of observed cases. We repeat this process 5 times, using an additional 4 week's worth of data each time (adding approximately one more month for each case). In particular, we take the following 5 lengths (52, 56, 60, 64, and 68) to be the number of weeks of incidence case data included in our partial data vectors, and run the predictive model using each of these partial data lengths. We compare predicted results to observed results in all cases.

Table (2.3) Measles parameters

Parameter	Value	Definition
$1/\kappa$	7.5/7	Average incubation period
$1/\gamma$	6.5/7	Average time from symptom onset to recovery
σ	0	Average time for the loss of immunity
μ	1/3120	Birth/death rate
α	0.6	Scaling component

For any partial data length, we let incidence case vector D be the partial data vector of weekly measles incidence cases described in Section 2.3.1. We aim to solve (2.12) by discretizing $\beta(t)$ using two different bases, the basis of trigonometric functions described in Section 2.1.2 and the basis of Legendre polynomials described in Section 2.1.2.

To initialize the iterative process for each forecast, we start as before, with an initial guess A_0 for the coefficients of $\hat{\beta}(A,t)$. Since we have no a priori knowledge about the transmission rate, and to avoid confirmation bias given that there is a discernable pattern of spikes in observed cases, we randomly select a constant λ from the uniform distribution on $[3,4]$ to serve as the initial (constant) guess for transmission rate $\hat{\beta}_0 = \hat{\beta}(A_0,t)$. Then the expansion coefficients A_0 are chosen so $\hat{\beta}_0 \approx \lambda$ for all $t \in [1,b]$, where b is the length of D.

We set the initial conditions $S_0 - R_0$ in system (2.7) - (2.10) as follows

$$S(0) = N - I(0), \quad E(0) = 0, \quad I(0) = 240, \quad R(0) = 0$$

where N is given in Section 2.3.1 and $I(0)$ is selected to be the first datapoint in D. That is, $D(1) = 240$ is the first week's recorded number of incidence cases during this time frame. We fix the system parameters as defined in Table 2.3, according to the observations made by Anderson and May, as recorded in [66]. We further note that $\sigma = 0$ because measles immunity is permanent, and α is a hand-tuned parameter selected by experimentation to recover accurate projections.

Incidence case recovery. We test our method by projecting $\beta(t)$ onto the basis of trigonometric functions defined in Section 2.1.2. We set the number of basis pairs to $\overline{N} = 10$, yielding a basis of dimension $2\overline{N} + 1 = 21$. We set $L = 3$ so that the mean of recovered transmission rates have an approximate periodicity of 52 weeks, coinciding with a priori knowledge of the timing of cyclic spikes in measles incidence cases. We set the singular value truncation threshold to $\sqrt{\epsilon} = 3.5$ which, as for the case with simulated data, is hand-tuned to reduce the amount of overfitting resulting from having noisy data. The step size for (2.13) is set to $\psi = 0.1$ and we run the loop for 60 iterations. For each partial data vector D, we sample $\hat{\beta}_0$ 100 times, running (2.13) for each sample. As was the case for simulated data, we generate a new random noisy incidence case vector to use as the input data in the minimization problem. To generate the noisy data, we add noise to each datapoint of the base incidence case vector D. The noise at each point is selected according to the Poisson distribution with parameter proportional to the number of cases at that point. More specifically, if the number of observed cases during week 5 is 500, (so that $D(5) = 500$), we generate random noise by using MATLAB's poissrnd() function as follows:

```
poissrnd(2*500) - poissrnd(2*500).
```

In Figure 2.17 we see a plot of the transmission rates recovered with 52 and 68 weeks of partial data from the London measles epidemic, respectively. From inspecting the plot, we see that the transmission rate during the recorded spike in incidence cases is more accurately recovered as a higher value when more data during this spike was used. In particular, we see in Figure 2.16 that the partial data vector of length 68 includes the peak of the second spike observed during this time frame. We conclude that the transmission rate recovered using 68 weeks of data is improved over the rate recovered using 52 weeks of data due to the inclusion of this spike in the incidence case data.

Next we plot the recovered and projected incidence cases using both 52 and 68 weeks of data respectively. The results are shown in Figures 2.18 - 2.19. As expected, we see that the incidence cases recovered using a longer partial data vector D produce a more accurate recovered incidence case data vector when compared to the observed cases recorded

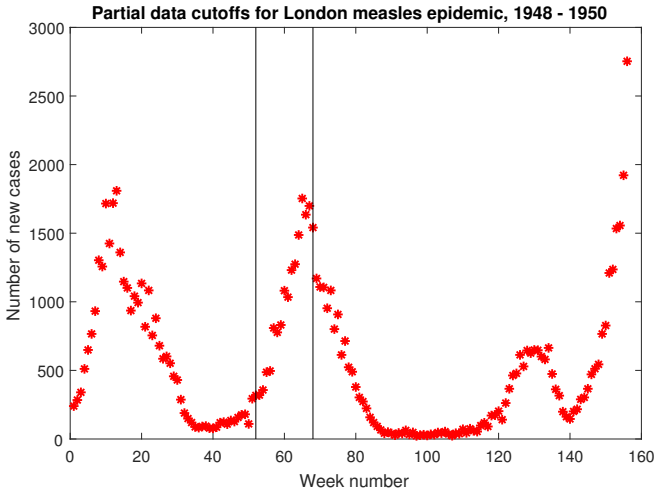

Figure (2.16) Partial data cutoff bounds, London measles epidemic

The cutoffs are shown as thin black vertical bars. The left vertical bar indicates the cutoff point for a partial data vector of length 52, while the right vertical bar indicates the cutoff point for a partial data vector of length 68.

in the data. The projected incidence cases are also more accurate in near term projections (projections for the weeks immediately following the last week of data used in a partial data vector) when longer partial data vectors are used. By referring to Figure 2.20, one sees these results plotted against recorded measles cases.

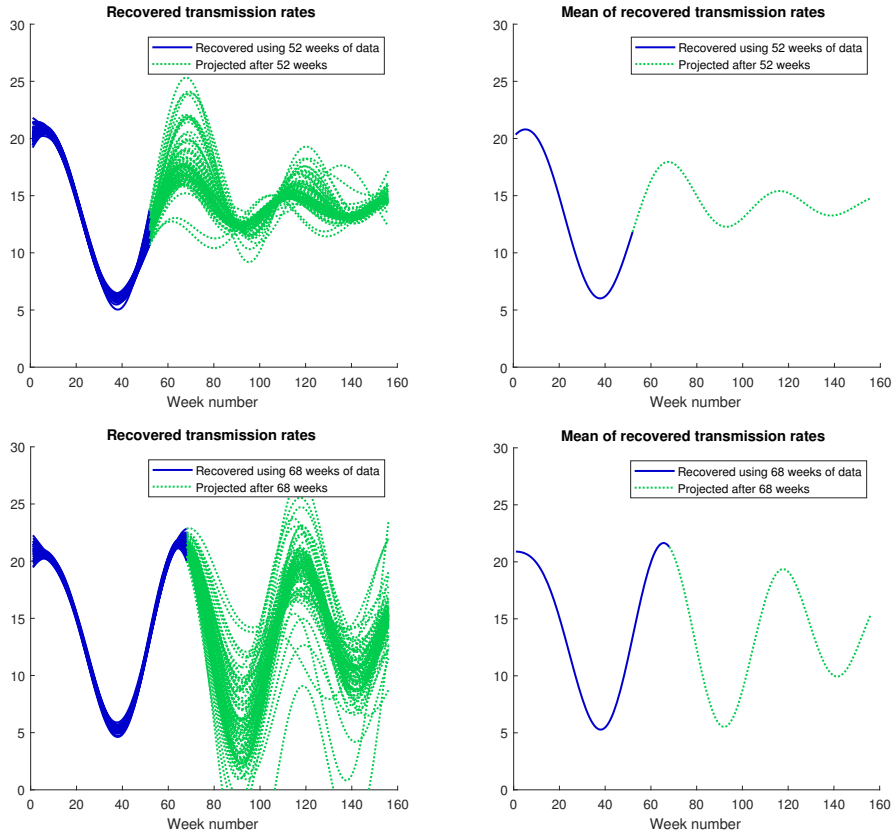

Figure (2.17) London, 1948-1950. Recovered transmission rates and projections, bundles and mean.

Left: Each recovered $\hat{\beta}_K$. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the projected transmission rate (the rate where data was not available). Right: The mean of all $\hat{\beta}_K$, with solid and dotted lines indicating recovered rates and projected rates, respectively.

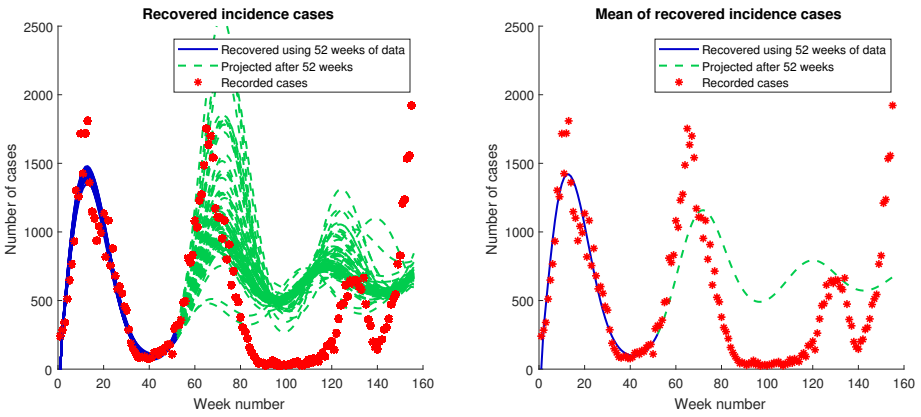

Figure (2.18) London measles incidence cases recovered using 52 weeks of data.

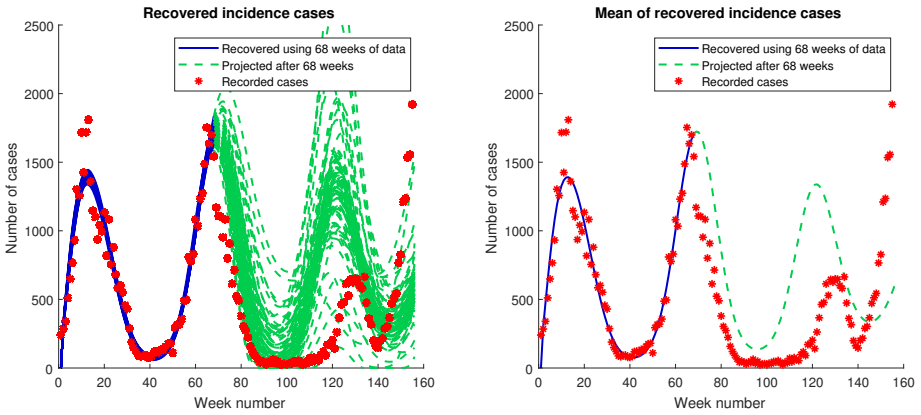

Figure (2.19) London measles incidence cases recovered using 68 weeks of data.

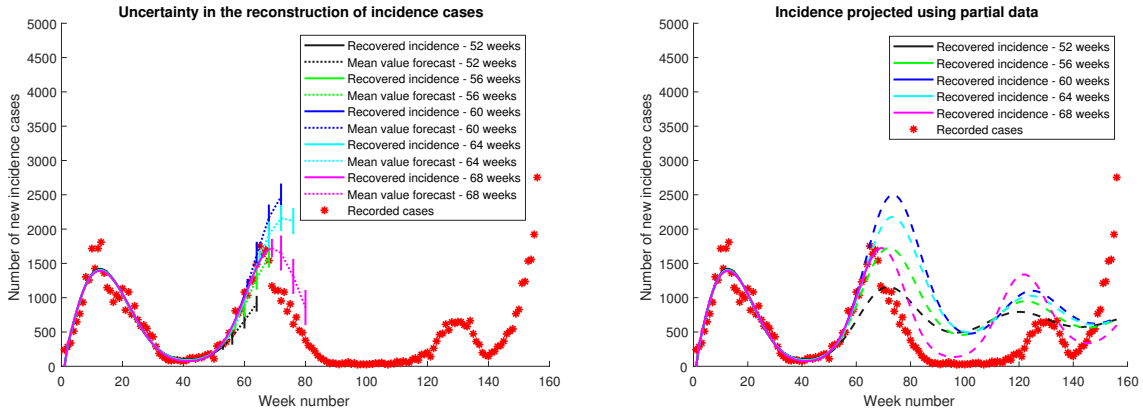

Figure (2.20) London measles incidence cases, projections and mean value forecasts.

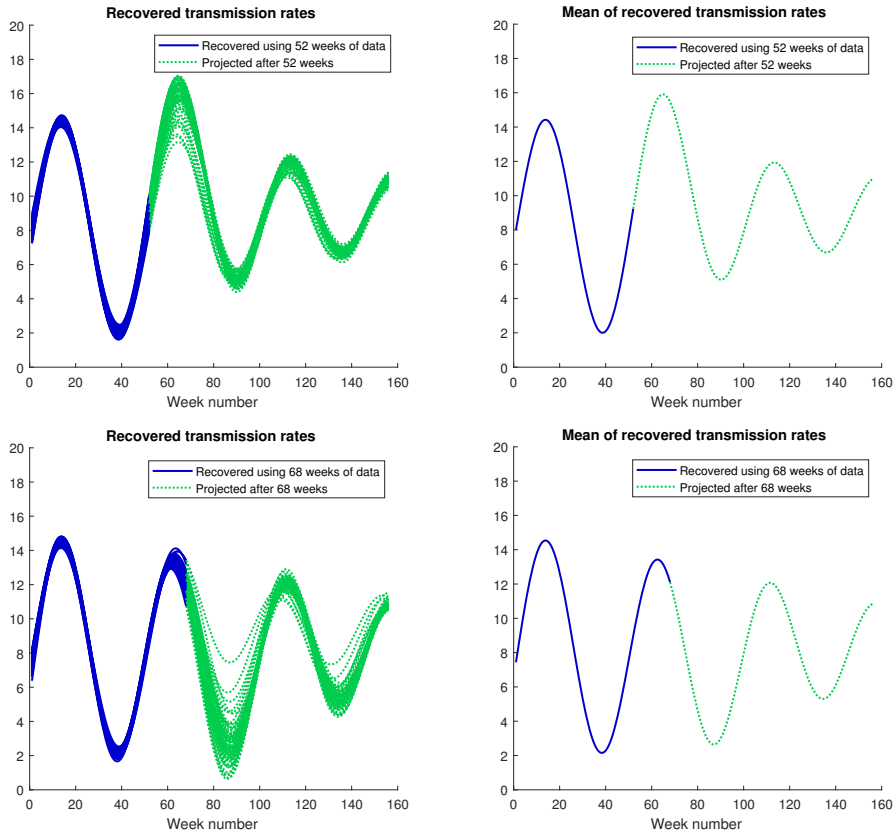

Figure (2.21) Birmingham, 1948-1950. Recovered transmission rates and projections, bundles and mean.

Left: Each recovered $\hat{\beta}_K$. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the projected transmission rate (the rate where data was not available). Right: The mean of all $\hat{\beta}_K$, with solid and dotted lines indicating recovered rates and projected rates, respectively.

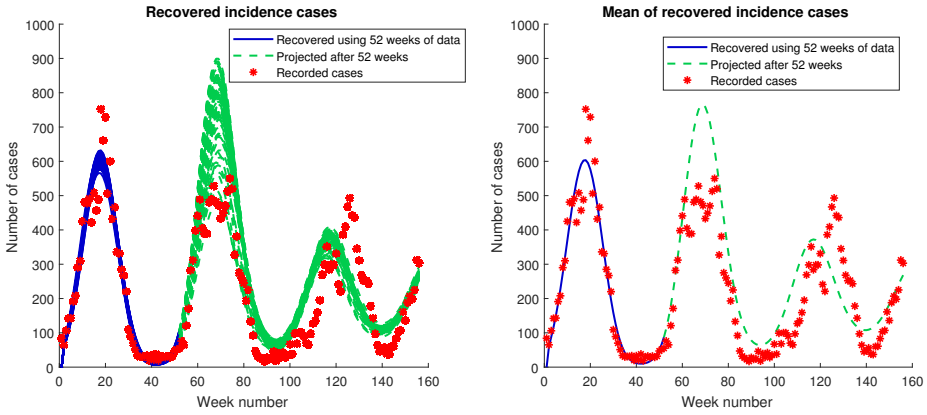

Figure (2.22) Birmingham measles incidence cases recovered using 52 weeks of data.

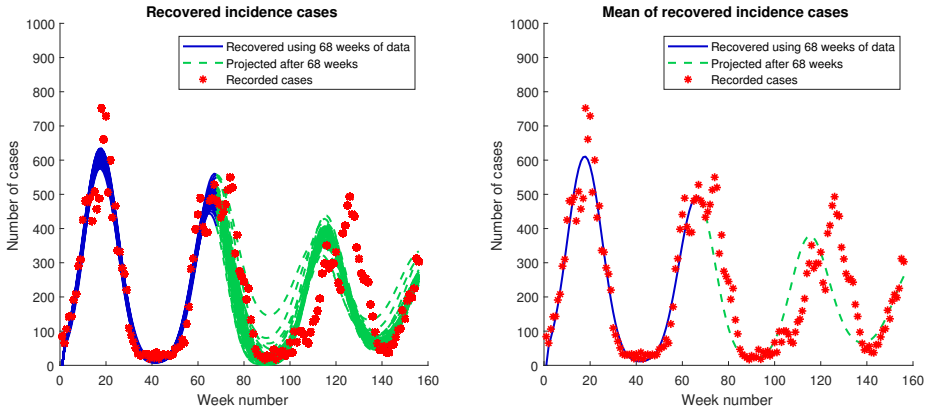

Figure (2.23) Birmingham measles incidence cases recovered using 68 weeks of data.

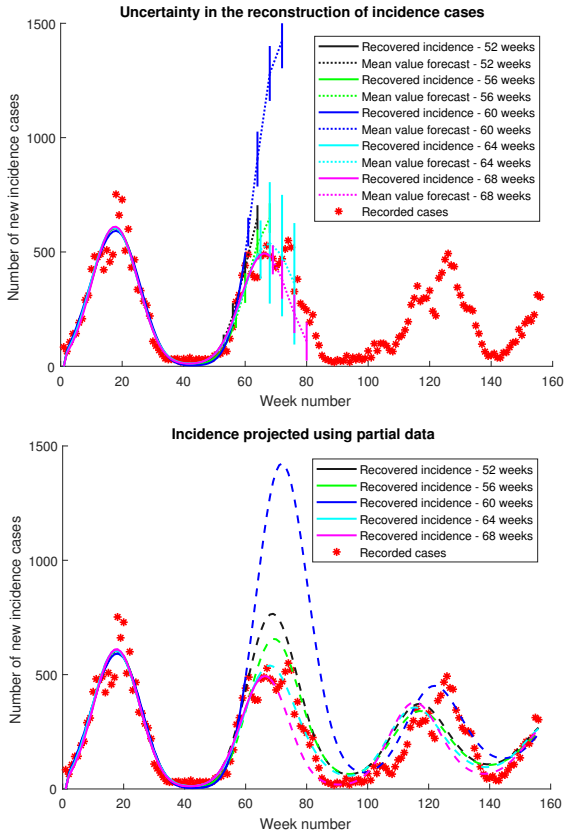

Figure (2.24) Birmingham measles incidence cases, projections and mean value forecasts.

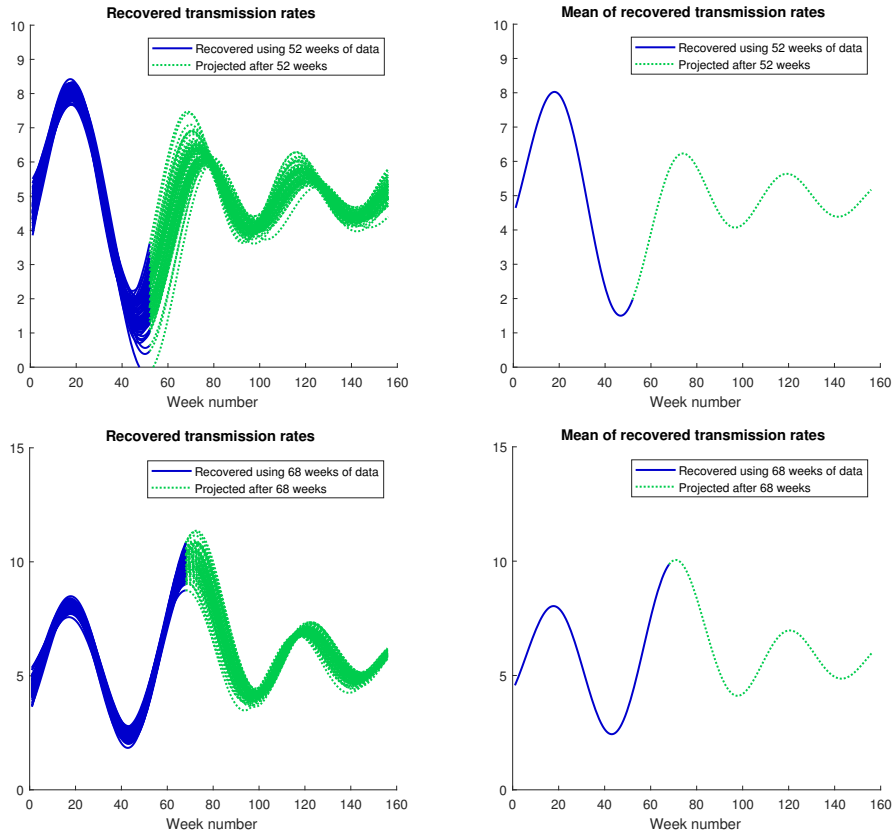

Figure (2.25) Newcastle, 1948-1950. Recovered transmission rates and projections, bundles and mean.

Left: Each recovered $\hat{\beta}_K$. The solid lines indicate values where partial data was used for recovery, while the dotted lines show the projected transmission rate (the rate where data was not available). Right: The mean of all $\hat{\beta}_K$, with solid and dotted lines indicating recovered rates and projected rates, respectively.

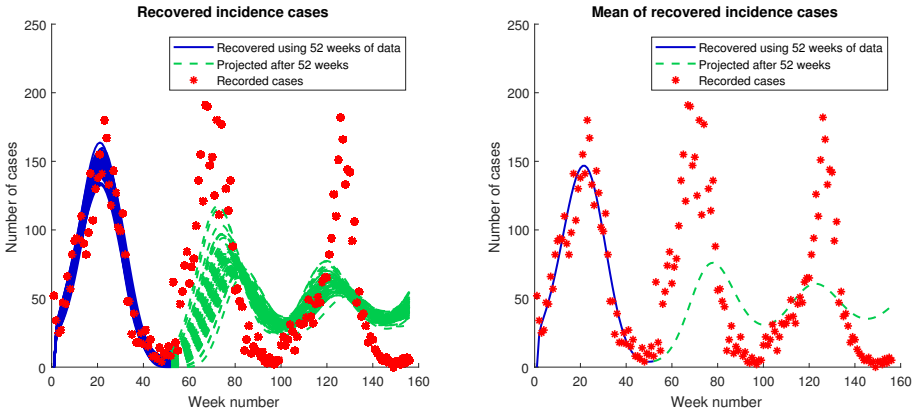

Figure (2.26) Newcastle measles incidence cases recovered using 52 weeks of data.

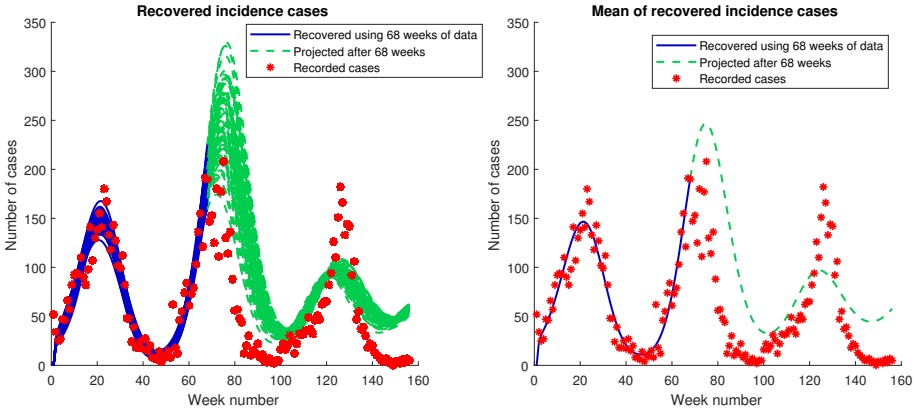

Figure (2.27) Newcastle measles incidence cases recovered using 68 weeks of data.

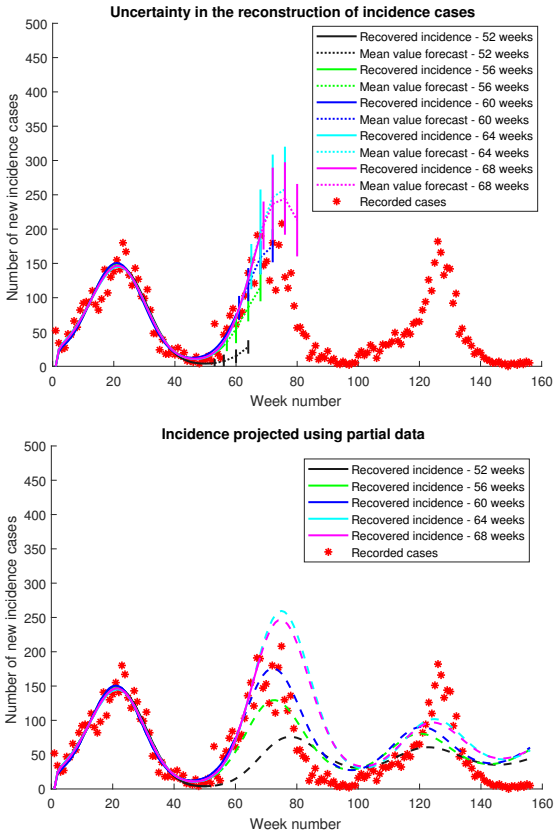

Figure (2.28) Newcastle measles incidence cases, projections and mean value forecasts.

PART 3

DECENTRALIZED PARAMETER ESTIMATION

3.0.1 Introduction

Now we turn our attention to applying decentralization concepts to parameter estimation problems in epidemiology. The scenario we consider is one in which different regions/cities/nations ("nodes") are experiencing outbreaks of the same disease, and each participating node desires to perform transmission rate recovery and incidence case forecasting individually. We set out to determine whether collaboration between nodes is feasible, and whether collaboration can yield helpful results in recovering past incidence case data and a common transmission rate.

In this scenario, each node privately holds its individual incidence case data with the goal of recovering a transmission rate that is common across all regions. This decentralization approach has the benefit of helping all nodes ascertain an estimate for their disease transmission rate while not giving away their incidence case data, which may be beneficial in cases where specific outbreak numbers may be considered sensitive. Therefore, we explore whether this extension of decentralization can help nodes collaborate on achieving their common objective of transmission rate recovery while witholding trust (by not granting outsiders knowledge of outbreak details).

Building on our previous work, we consider the same SEIR compartmental model of epidemiology in our approach, working under the assumption that the disease in question is cyclic in nature. As before, we work under the assumption of a cyclic transmission rate $\beta(t)$, and we consider the same system of ODE's as in our previous work. In the decentralized case, each participating node maintains and solves its own system of ODE's with its own estimates for the parameters $\kappa, \gamma, \sigma, \mu$ and α.

To motivate the extension of 2.13 to the decentralized case, we build upon the work in [64] by showing that the error in the centralized, single-node case scales linearly with the number of participating regions in the decentralized case, and such error may be mitigated by choosing tighter parameters on the error tolerance for each node or by increasing the number of iterations in the stopping criteria.

In what follows, the results and analysis closely follow from [64] under small modifications to the iterative scheme, allowing the result to be extended to the decentralized, consensus case. We are interested in solving the nonlinear operator equation

$$\sum_{i=1}^{C} F_i(A) = 0,\ F_i : \mathcal{D}_F \subset \mathcal{X} \to \mathcal{Y} \tag{3.1}$$

at each node i, where $\hat{A}$ is a solution of interest. We consider the general form

$$A_i^{(k+1)} = \tilde{A}_i^{(k)} + \psi\big[(I - P_i^{(k)})(\xi_i^{(k-1)} - A_i^{(k)}) - Q_i^{(k)} F_i(A_i^{(k)})\big] \tag{3.2}$$

$$J_i^{(k+1)} = J_i^{(k)} + \frac{\langle s_i^{(k)}, \cdot \rangle}{\|s_i^{(k)}\|^2} \left(y_i^{(k)} - J_i^{(k)} s_i^{(k)}\right) \tag{3.3}$$

where $\psi > 0$ is the step-size, and $\tilde{A}_i^{(k)} = \sum_{j=1}^{C} w_{ij} A_j^{(k)}$ for all i, where $\sum_{j=1}^{C} w_{ij} = 1$. We show that this modification at each node – using the convex combination of solution estimates from neighbor nodes in each iteration, rather than a single solution estimate – does not impact the convergence results as established in [64].

Theorem 7. *Suppose the following conditions are fulfilled:*

1. *F_i is Fréchet differentiable in a neighborhood of $\hat{A}$ with F_i' being a compact Lipschitz-continuous operator with Lipschitz constant L_i.*

2. *$A_i^{(0)} \in \mathcal{D}_F \subset \mathcal{X}$ and $J_i^{(0)}$ are node i's initial approximations for $\hat{A}$ and $F_i'(\hat{A})$, respectively.*

3. *The sequences $\{A_i^{(k)}\}$ and $\{J_i^{(k)}\}$ are generated at each node according to* (3.2) *and* (3.3)*, respectively.*

4. *For all i, $\|A_i^{(0)} - \hat{A}\| \leq R$, $\|J_i^{(0)} - G_i^{(0)}\| \leq \delta_i$, and $\{\xi_i^{(k)}\}$ are chosen to satisfy $\psi\|(I - P_i^{(k)})(\xi_i^{(k)} - \hat{A})\| \leq \varphi$ for all k and for some φ satisfying $q^{\hat{k}_i} R \leq \frac{\varphi}{1-q}$, where $\hat{k}_i$ is the iteration at node i for which the inequality holds for the first time. Denote $G_i^{(k)} = \int_0^1 F_i'(A_i^{(k)} + t(\hat{A} - A_i^{(k)}))dt$ for all i. In addition, suppose R satisfies*

$$R \leq \frac{2(1-q)\delta}{L(5q+1)} \text{ where } \delta = \max_i \delta_i, L = \max_i L_i, \text{ and} \tag{3.4}$$

$$q = 1 - \psi\left(1 - \frac{2\delta}{\sqrt{\epsilon}}\right) \text{ where } \epsilon = \min_i \epsilon_i. \tag{3.5}$$

where ϵ_i is the singular value truncation parameter at node i.

Then, terminating the method at iteration $\hat{k}_i$ yields the accuracy

$$\|A_i^{\hat{k}_i} - \hat{A}\| \leq \frac{2\varphi}{1-q}.$$

Proof. First we observe that we want $q \in (0, 1)$; i.e. we want to ensure that our choice of $\psi, \delta,$ and ϵ allow for:

$$0 < 1 - \psi\left(1 - \frac{2\delta}{\sqrt{\epsilon}}\right) < 1 \tag{3.6}$$

which is equivalent to

$$\frac{\psi - 1}{\psi} < \frac{2\delta}{\sqrt{\epsilon}} < 1.$$

Noting that $\delta, \epsilon > 0$ we see that $2\delta/\sqrt{\epsilon} > 0$ so the lower bound in (3.6) holds automatically independent of the choice for ψ. Also, we see that $2\delta/\sqrt{\epsilon} < 1 \iff \delta < \sqrt{\epsilon}/2$, and truncation parameters ϵ_i may be chosen across all i to satisfy this inequality based on the accuracy of the initial approximation $G_i^{(0)}$.

To continue, we proceed as in [64] by using induction to show that for $1 \leq k \leq \hat{k}$ we have

$$\|J_i^{(k)} - G_i^{(k)}\| \leq (2 - q^k)\delta. \tag{3.7}$$

and $A_i^{(k+1)}$ satisfying

$$\|A_i^{(k+1)} - \hat{A}\| \leq q^{k+1}R + \varphi \sum_{j=0}^{k} q^j. \tag{3.8}$$

Proving (3.7) for $k = 0$ follows immediately by assumption: $\|J_i^{(0)} - G_i^{(0)}\| \leq \delta = (2 - q^0)\delta$. The proof of the bound (3.8) for $k = 0$ is the same as for the inductive step, so it is omitted.

Now suppose $k = n < \hat{k}$. From (3.2)-(3.3), for any i we have

$$\begin{aligned}
J_i^{(n)} - G_i^{(n)} &= \left[J_i^{(n-1)} + \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} \left(y_i^{(n-1)} - J_i^{(n-1)} s_i^{(n-1)} \right) \right] - G_i^{(n)} \\
&= \left[J_i^{(n-1)} + \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} y_i^{(n-1)} - \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} J_i^{(n-1)} s_i^{(n-1)} + G_i^{(n-1)} \right. \\
&\quad \left. - G_i^{(n-1)} + \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} G_i^{(n-1)} s_i^{(n-1)} - \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} G_i^{(n-1)} s_i^{(n-1)} \right] - G_i^{(n)} \\
&= J_i^{(n-1)} - \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} J_i^{(n-1)} s_i^{(n-1)} - G_i^{(n-1)} + \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} G_i^{(n-1)} s_i^{(n-1)} \\
&\quad + \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} y_i^{(n-1)} - \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} G_i^{(n-1)} s_i^{(n-1)} + G_i^{(n-1)} - G_i^{(n)} \\
&= \left(J_i^{(n-1)} - G_i^{(n-1)} \right) \left(I - \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} s_i^{(n-1)} \right) \\
&\quad + \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} \left(y_i^{(n-1)} - G_i^{(n-1)} s_i^{(n-1)} \right) + G_i^{(n-1)} - G_i^{(n)}
\end{aligned} \tag{3.9}$$

where $s_i^{(n-1)} = A_i^{(n)} - A_i^{(n-1)}$ and $y_i^{(n)} = F_i(A_i^{(n+1)}) - F_i(A_i^{(n)})$.

By definitions of $y_i^{(n)}$ and $F_i'(A_i^{(n)})$, we have

$$\begin{aligned}
&y_i^{(n-1)} - G_i^{(n-1)} s_i^{(n-1)} \\
&= \int_0^1 \left(F_i'(A_i^{(n-1)} + t(A_i^{(n)} - A_i^{(n-1)})) - F_i'(A_i^{(n-1)} + t(\hat{A} - A_i^{(n-1)})) \right) dt \cdot (A_i^{(n)} - A_i^{(n-1)})
\end{aligned}$$

By the Lipschitz continuity of F_i', for any $h \in \mathcal{D}_F \subset \mathcal{X}$ we have

$$\begin{aligned}
&\left\| \frac{\langle s_i^{(n-1)}, h \rangle}{\|s_i^{(n-1)}\|^2} \left(y_i^{(n-1)} - G_i^{(n-1)} s_i^{(n-1)} \right) \right\| \\
&\leq \frac{L\|h\|}{\|s_i^{(n-1)}\|} \int_0^1 t \|\hat{A} - A_i^{(n)}\| dt \|s_i^{(n-1)}\| \leq \frac{L\|h\|}{2} \|\hat{A} - A_i^{(n)}\|. \qquad (3.10)
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&\|G_i^{(n-1)} - G_i^{(n)}\| \\
&= \left\| \int_0^1 \left(F_i'(A_i^{(n-1)} + t(\hat{A} - A_i^{(n-1)})) - F_i'(A_i^{(n)} + t(\hat{A} - A_i^{(n)})) \right) dt \right\| \\
&\leq \frac{L}{2} \|A_i^{(n)} - A_i^{(n-1)}\| \\
&\leq \frac{L}{2} \left(\|A_i^{(n)} - \hat{A}\| + \|\hat{A} - A_i^{(n-1)}\| \right) \qquad (3.11)
\end{aligned}$$

Combining (3.10) and (3.11), and by observing that the norm of the orthogonal projector $\left[I - \frac{\langle s_i^{(n-1)}, \cdot \rangle}{\|s_i^{(n-1)}\|^2} s_i^{(n-1)} \right]$ is 1, we form the estimate

$$\left\| J_i^{(n)} - G_i^{(n)} \right\| \leq \left\| J_i^{(n-1)} - G_i^{(n-1)} \right\| + \frac{L}{2} \left(2\|A_i^{(n)} - \hat{A}\| + \|\hat{A} - A_i^{(n-1)}\| \right)$$

and by the induction assumptions, we arrive at

$$\left\| J_i^{(n)} - G_i^{(n)} \right\| \leq (2 - q^{n-1})\delta + \frac{L}{2} \left(2q^n R + q^{n-1} R + \frac{3\epsilon}{1-q} \right). \qquad (3.12)$$

Recalling that $\hat{k}_i$ is chosen to be the first iteration for which $q^{\hat{k}_i} R \leq \varphi/(1-q)$, we observe that if $n < \hat{k}_i$, then the reverse inequality necessarily holds; i.e. if $n < \hat{k}_i$, then $q^n R > \varphi/(1-q)$. Thus we have

$$\left\| J_i^{(n)} - G_i^{(n)} \right\| \leq (2 - q^{n-1})\delta + \frac{RLq^n}{2}(5 + q^{-1}). \qquad (3.13)$$

Using the estimate on R from (3.4) establishes bound (3.7):

$$\left\| J_i^{(n)} - G_i^{(n)} \right\| \leq (2 - q^{n-1})\delta + (1-q)\delta q^{n-1} = (2-q^n)\delta. \tag{3.14}$$

Next, we turn our attention to verifying (3.8) for $k = n$. From (3.2), for each node i we have

$$\begin{aligned} A_i^{(n+1)} - \hat{A} = \tilde{A}_i^{(n)} - \hat{A} + \psi \Big\{ & (I - P_i^{(n)})(\xi_i^{(n)} - A_i^{(n)}) - (I - P_i^{(n)})(A_i^{(n)} - \hat{A}) \\ & - Q_i^{(n)} J_i^{(n)} (A_i^{(n)} - \hat{A}) + Q_i^{(n)} (J_i^{(n)} - G_i^{(n)})(A_i^{(n)} - \hat{A}) \Big\} \end{aligned}$$

From [64] we see that $Q^{(n)} J^{(n)} = P^{(n)}$ and so it follows that $Q_i^{(n)} J_i^{(n)} = P_i^{(n)}$ for all nodes i. Similarly, we have $\|Q_i^{(n)} z\|^2 \leq \frac{1}{\epsilon}\|z\|^2$ for any $z \in \mathcal{Y}$. Combining these results gives the bound

$$\begin{aligned} & \|A_i^{(n+1)} - \hat{A}\| \\ & \leq \|\tilde{A}_i^{(n)} - \hat{A} + \psi \left((I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - (I - P_i^{(n)})(A_i^{(n)} - \hat{A}) - P_i^{(n)}(A_i^{(n)} - \hat{A}) \right) \| \\ & \quad + \frac{\psi}{\sqrt{\epsilon}} \|(J_i^{(n)} - G_i^{(n)})(A_i^{(n)} - \hat{A})\| \\ & = \| \sum_{j=1}^{C} w_{ij}(A_j^{(n)} - \hat{A}) + \psi \left((I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - (I - P_i^{(n)})(A_i^{(n)} - \hat{A}) - P_i^{(n)}(A_i^{(n)} - \hat{A}) \right) \| \\ & \quad + \frac{\psi}{\sqrt{\epsilon}} \|(J_i^{(n)} - G_i^{(n)})(A_i^{(n)} - \hat{A})\| \\ & = \| \sum_{j \neq i} w_{ij}(A_j^{(n)} - \hat{A}) \\ & \quad + w_{ii}(A_i^{(n)} - \hat{A}) + \psi \left((I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - (I - P_i^{(n)})(A_i^{(n)} - \hat{A}) - P_i^{(n)}(A_i^{(n)} - \hat{A}) \right) \| \\ & \quad + \frac{\psi}{\sqrt{\epsilon}} \|(J_i^{(n)} - G_i^{(n)})(A_i^{(n)} - \hat{A})\| \\ & \leq \| \sum_{j \neq i} w_{ij}(A_j^{(n)} - \hat{A})\| \\ & \quad + \|w_{ii}(A_i^{(n)} - \hat{A}) + \psi \left((I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - (I - P_i^{(n)})(A_i^{(n)} - \hat{A}) - P_i^{(n)}(A_i^{(n)} - \hat{A}) \right) \| \\ & \quad + \frac{\psi}{\sqrt{\epsilon}} \|(J_i^{(n)} - G_i^{(n)})(A_i^{(n)} - \hat{A})\| \end{aligned} \tag{3.15}$$

Choose $\xi_i^{(n)} \in \mathcal{D}_F \subset \mathcal{X}$ and $\psi > 0$ so that $\psi\|(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A})\| \leq \varphi$. Combining this with (3.7) gives upper bounds for the last two terms of (3.15). More specifically, for the second term, we have

$$
\begin{aligned}
&\|w_{ii}(A_i^{(n)} - \hat{A}) + \psi\left((I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - (I - P_i^{(n)})(A_i^{(n)} - \hat{A}) - P_i^{(n)}(A_i^{(n)} - \hat{A})\right)\| \\
=&\|w_{ii}(A_i^{(n)} - \hat{A}) + \psi(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - \psi(I - P_i^{(n)})(A_i^{(n)} - \hat{A}) - \psi P_i^{(n)}(A_i^{(n)} - \hat{A})\| \\
=&\Big\|w_{ii}(A_i^{(n)} - \hat{A}) + \psi(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) \\
&- \psi(A_i^{(n)} - \hat{A}) + \psi P_i^{(n)}(A_i^{(n)} - \hat{A}) - \psi P_i^{(n)}(A_i^{(n)} - \hat{A})\Big\| \\
=&\|w_{ii}(A_i^{(n)} - \hat{A}) + \psi(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A}) - \psi(A_i^{(n)} - \hat{A})\| \\
=&\|w_{ii}(A_i^{(n)} - \hat{A}) - \psi(A_i^{(n)} - \hat{A}) + \psi(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A})\| \\
=&\|(w_{ii} - \psi)(A_i^{(n)} - \hat{A}) + \psi(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A})\| \\
\leq&(w_{ii} - \psi)\|(A_i^{(n)} - \hat{A})\| + \psi\|(I - P_i^{(n)})(\xi_i^{(n)} - \hat{A})\| \\
\leq&(1 - \psi)\|(A_i^{(n)} - \hat{A})\| + \varphi
\end{aligned}
\tag{3.16}
$$

where the last inequality is obtained by noting that $w_{ii} \leq 1$. For the last term of (3.15), we apply (3.7) to obtain

$$
\begin{aligned}
\frac{\psi}{\sqrt{\epsilon}}\|(J_i^{(n)} - G_i^{(n)})(A_i^{(n)} - \hat{A})\| &\leq \frac{\psi}{\sqrt{\epsilon}}\|J_i^{(n)} - G_i^{(n)}\|\|A_i^{(n)} - \hat{A}\| \\
&\leq \frac{\psi}{\sqrt{\epsilon}}(2 - q^n)\delta\|A_i^{(n)} - \hat{A}\|
\end{aligned}
\tag{3.17}
$$

Combining (3.16) and (3.17), we arrive at

$$
(1 - \psi)\|A_i^{(n)} - \hat{A}\| + \psi\frac{2 - q^n}{\sqrt{\epsilon}}\delta\|A_i^{(n)} - \hat{A}\| + \varphi \tag{3.18}
$$

and by definition of q, we form the following estimate for (3.18):

$$(1-\psi)\|A_i^{(n)}-\hat{A}\|+\psi\frac{2-q^n}{\sqrt{\epsilon}}\delta\|A_i^{(n)}-\hat{A}\|+\varphi\leq\left(1-\psi\left(1-\frac{2\delta}{\sqrt{\epsilon}}\right)\right)\|A_i^{(n)}-\hat{A}\|+\varphi$$
$$=q\|A_i^{(n)}-\hat{A}\|+\varphi. \tag{3.19}$$

For the first term of (3.15) we can apply the inductive step and we arrive at

$$\|\sum_{j\neq i}w_{ij}(A_j^{(n)}-\hat{A})\|\leq\sum_{j\neq i}w_{ij}\|A_j^{(n)}-\hat{A}\|\leq q^nR+\varphi\sum_{j=0}^{n}q^j \tag{3.20}$$

Applying the inductive assumption to (3.19) we arrive at the bound:

$$q\|A_i^{(n)}-\hat{A}\|+\varphi\leq q^{n+1}R+\varphi\sum_{j=0}^{n}q^j. \tag{3.21}$$

Combining (3.20) and (3.21) gives the bound

$$\|A_i^{(n+1)}-\hat{A}\|\leq 2(q^nR+\varphi\sum_{j=1}^{n}q^j). \tag{3.22}$$

We complete the proof by observing that the second term of the non-constant factor in (3.22) is a geometric sum, and by using bound (3.4) for the factor of R. □

It is important to consider when the choices of $\xi^{(k)}$, ψ, and φ guarantee a higher accuracy of the approximate solution as compared with the accuracy of the initial approximation. One observes ([64]) that the accuracy of the approximate solution $A_i^{(\hat{k}_i)}$ is higher when the test function $\xi_i^{(k)}$ is close to $\hat{A}$ in terms of structure. More specifically, this is achieved when either

$$\xi_i^{(k)}-\hat{A}=\left(J_i^{(k)}\right)^*J_i^{(k)}w_i^{(k)}+\eta_i^{(k)},\ \ w^{(k)},\eta^{(k)}\in\mathcal{D}_F\subset\mathcal{X} \tag{3.23}$$

with $(I-P_i^{(k)})\eta_i^{(k)}=0$, or when $\|(I-P_i^{(k)})\eta_i^{(k)}\|\leq\varrho$ for some negligible ϱ.

The result from Theorem 7 establishes that using the convex combination of solution estimates does not alter the convergence results established in [64]. One observes that by summing the error over all nodes, the global error scales linearly with the number of nodes participating. Also, by defining $A_i^{(k+1)}$ as the sum $\tilde{A}_i^{(k+1)} + \psi\Big(\ \cdot\ \Big)$, one sees that choosing a smaller value of ψ leads to smaller error in the deviation of an individual node's solution estimate to the consensus solution. We conclude by mentioning that one way of choosing $\{\xi_i^{(k)}\}$ satisfying the assumptions of Theorem 7 is to define $\xi_i^{(k)} = A_i^{(k-1)}$ for all $k \geq 1$. This is our approach when implementing the method for numerical experimentation.

3.0.2 Numerical approach

As we have shown, our numerical approach for the decentralized case is similar to our previous work, with minor modifications made where necessary and appropriate to extend this application in a decentralized manner. As before, to implement our strategy numerically, we aim to recover a vector of coefficients which are used to form a linear combination of basis elements in estimating the true transmission rate $\beta(t)$. For nodes to collaborate in solving for a common coefficient vector, the basis elements must be known and shared among participating nodes. This forms one of the critical assumptions of our method: it is necessary for participants to declare and use a common basis to allow for collaborative transmission rate recovery between nodes. In this spirit, if each node i maintains an estimate A_i of the coefficient vector, we define the consensus coefficient vector $\overline{A} = \sum_{i=1}^{C} A_i$ where C is the number of nodes participating in the transmission rate recovery. Then our goal of recovering a common transmission rate across all nodes is equivalent to the goal of having each node's coefficient vector estimate approach (or be acceptably close to) the consensus coefficient vector.

Suppose $\big[S_i(A_i,t), E_i(A_i,t), I_i(A_i,t), R_i(A_i,t)\big]$ is node i's numerical solution to the system

$$\frac{dS_i}{dt} = \mu_i N_i - \hat{\beta}(A_i,t)S_i(t)\frac{I_i^{\alpha_i}(t)}{N_i} - \mu_i S_i(t) + \sigma_i R_i(t) \tag{3.24}$$

$$\frac{dE_i}{dt} = \hat{\beta}(A_i,t)S_i(t)\frac{I_i^{\alpha_i}(t)}{N_i} - \mu_i E_i(t) - \kappa_i E_i(t) \tag{3.25}$$

$$\frac{dI_i}{dt} = \kappa_i E_i(t) - \gamma_i I_i(t) - \mu_i I_i(t) \tag{3.26}$$

$$\frac{dR_i}{dt} = \gamma_i I_i(t) - \mu_i R_i(t) - \sigma_i R_i(t) \tag{3.27}$$

$$S_i(0) = S_{i,0}, \quad E_i(0) = E_{i,0}, \quad I_i(0) = I_{i,0}, \quad R_i(0) = R_{i,0}. \tag{3.28}$$

Given node i's local incidence case data D_i, each node considers the unconstrained least squares minimization problem

$$\min_A \frac{1}{2}\,||\Phi_i(A) - D_i||^2 \tag{3.29}$$

where $\Phi_i(A) = \kappa_i E_i(A)$. To recover the vector of expansion coefficients A, each node solves (3.29) iteratively, using the regularized version of Broyden's secant method modified for decentralied use:

$$\begin{aligned} A_i^{k+1} &= \tilde{A}_i^k + \psi\big[(I - P_i^k)(A_i^{k-1} - A_i^k) - Q_i^k F_i(A_i^k)\big] \\ J_i^{k+1} &= J_i^k + \frac{\langle s_i^k, \cdot\rangle}{\|s_i^k\|^2}\left(y_i^k - J_i^k s_i^k\right) \end{aligned} \tag{3.30}$$

where $P_i^k, Q_i^k, y_i^k, s_i^k, J_i^k$ and F_i^k are the node i analogues of the method considered previously.

3.1 Testing the method

3.1.1 Simulating a model outbreak

We test our method by simulating 7 outbreaks across 7 regions (one outbreak per region) of a cyclic disease common to each outbreak. To simulate the outbreak, we define our model transmission rate to be $\beta(t) = 10\left(\sin(\pi t/26) + 1\right)\cdot \exp(t/500)$. We choose the coefficients

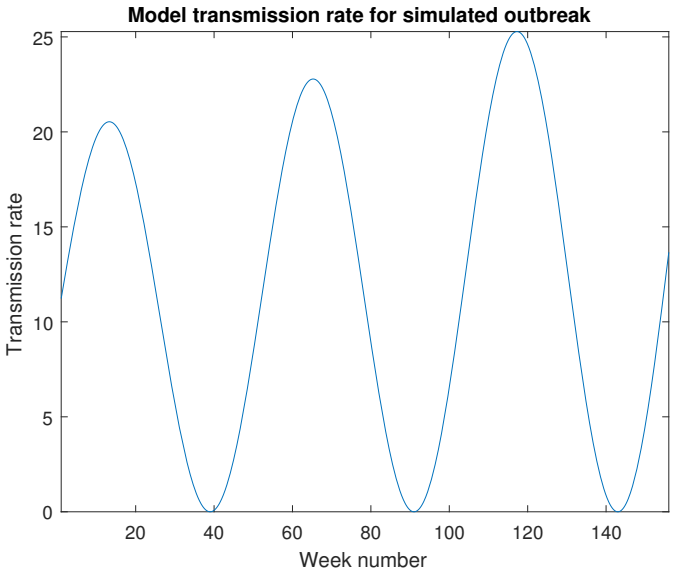

Figure (3.1) Objective transmission rate to be recovered.

for the argument of sine to model a disease that exhibits an annual spike in incidence cases (here we are measuring weekly incidence case data), while including the exponential factor to account for population growth over time. A plot of the model transmission rate is shown in Figure 3.1.

We determine the initial conditions as follows. For each node i, we choose $S_{i,0}$ to be an integer selected uniformly randomly from $[475000, 1475000]$ so that $475,000 \leq S_{i,0} \leq 1,475,000$ for all i. We choose $E_{i,0}$ to be an integer selected uniformly randomly from the interval $[100, 200]$, and we set $I_{i,0} = 0$ and $R_{i,0} = 0$ for all i. Once these initial conditions are generated, we set the population N_i for each region to be the sum $N_i = S_{i,0} + E_{i,0} + I_{i,0} + R_{i,0}$. Aiming to simulate a measles outbreak in each region, we generate our parameters according to a priori knowledge of the mean values for system parameters. At each node, we choose μ_i so that $\frac{1}{3100} \leq \mu_i \leq \frac{1}{3140}$ (uniformly random). Furthermore, suppose that for each i, $\epsilon_{i,1}$ and $\epsilon_{i,2}$ are selected randomly accoring to the standard uniform distribution. Then we choose $\kappa_i = 7/(7 + \epsilon_{i,1})$ and $\gamma_i = 7/(6 + \epsilon_{i,2})$. We also choose scaling parameter α_i accoring to the uniform distribution on $[0.5, 0.6]$ for each node. Lastly, we let $\sigma_i = 0$ for all nodes since we know that measles immunity is permanent. We show the generated populations and initial conditions for each region in Table 3.1, and the parameters for each region in Table 3.2.

Table (3.1) Population and initial conditions by region

Region	N_i	$S_{i,0}$	$E_{i,0}$	$I_{i,0}$	$R_{i,0}$
1	827,784	827,637	147	0	0
2	764,000	763,851	149	0	0
3	1,207,375	1,207,238	137	0	0
4	1,370,105	1,369,922	183	0	0
5	886,435	886,257	178	0	0
6	1,397,224	1,397,100	124	0	0
7	1,340,041	1,337,853	188	0	0

With the selected transmission rate, system parameters, and initial conditions, we set the simulated outbreak length to $m = 156 (= b - a)$ and we run MATLAB's ode23s ODE solver to simulate incidence case data in each region as before, adding noise to simulate real world inaccuracy in incidence case data collection. We plot the simulated outbreaks for all regions in Figure 3.2.

Table (3.2) System parameters by region

Region	κ_i	γ_i	α_i	μ_i
1	0.9351	1.0256	0.5714	3.2113e-04
2	0.9301	1.0842	0.5190	3.1949e-04
3	0.9450	1.0243	0.5276	3.1857e-04
4	0.9497	1.0601	0.5084	3.1959e-04
5	0.9402	1.1031	0.5925	3.2010e-04
6	0.9763	1.1323	0.5998	3.2020e-04
7	0.8819	1.0215	0.5972	3.2154e-04

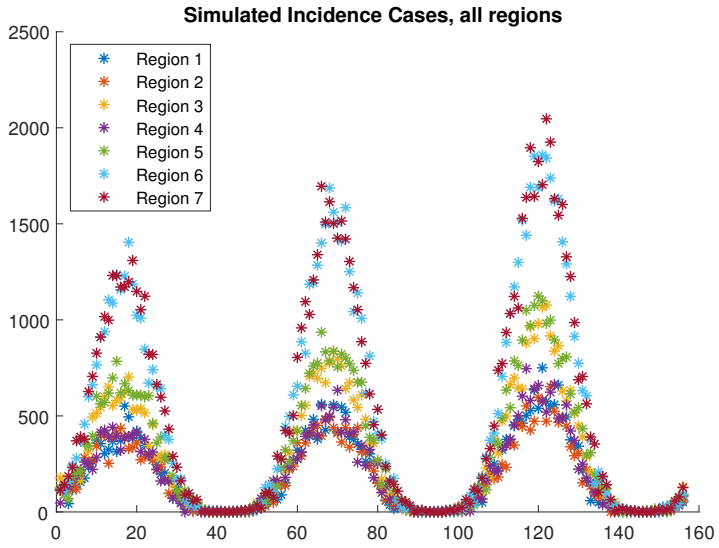

Figure (3.2) Simulated incidence case data, all regions.

3.1.2 Recovering a common transmission rate

Once the outbreaks are simulated, we assume that each node stores its own respective incidence case data vector, and does not share it with other nodes. We then aim to test the feasibility of the method by checking the accuracy of the consensus transmission rate when each node uses its entire observed incidence case data for recovery. To proceed, we define the common basis to be $\{s_n(u(t)), c_n(u(t)) : 1 \leq N \leq 14\} \cup \{1\}$ where $s_n(u(t)) = \sin(2\pi n \cdot u(t)/L)$ and $c_n(u(t)) = \cos(2\pi n \cdot u(t)/L)$ for $L = 3$. As before, we linearly interpolate $t \in [1, 156]$ to $u(t) \in [0, 1]$ to maintain consistency across all epidemic lengths. Once the basis

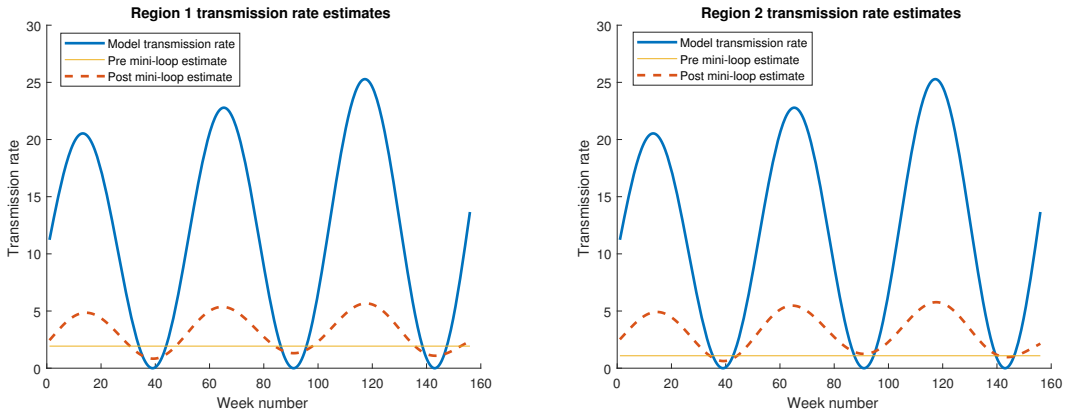

Figure (3.3) Comparing the preliminary transmission rate estimate with the result after the mini-loop.

is fixed, we assume each node independently estimates the initial transmission rate to be a constant selected uniformly randomly from the interval $[1, 3]$, and each node i determines a preliminary initial coefficient vector $\hat{A}_i^0$. To improve accuracy and speed of transmission rate and incidence case recovery, we perform an initial mini-loop of (3.30), starting with $\hat{A}_i^0$ and looping for 20 iterations, yielding a non-constant estimate of $\beta_0(t)$ at each node. A comparison of initial guess vs. preliminary transmission rate estimate β_0 is shown for two nodes in Figure 3.3. Once each node determines the coefficients for its privately held $\beta_0(t)$ estimate, this becomes the initializing coefficient vector A_i^0 in (3.30).

Communication is assumed to occur over the internet, so geography/range is not a factor in limiting the connectivity of 2 nodes. As such, it is assumed that all participating nodes are interconnected, so that the mixing matrix W has no zero entries (it should be noted that this is not a necessary requirement and our previous work in pure decentralized optimization references the fact that the mixing matrix need only satisfy double stochasticity and positive semi-definiteness). For the purposes of this simulation, we let $W = (I + J/7)/2$, where J is the all 1's matrix. We utilize a step-size $\psi = \frac{1}{10}$ and then run (3.30) for 250 iterations.

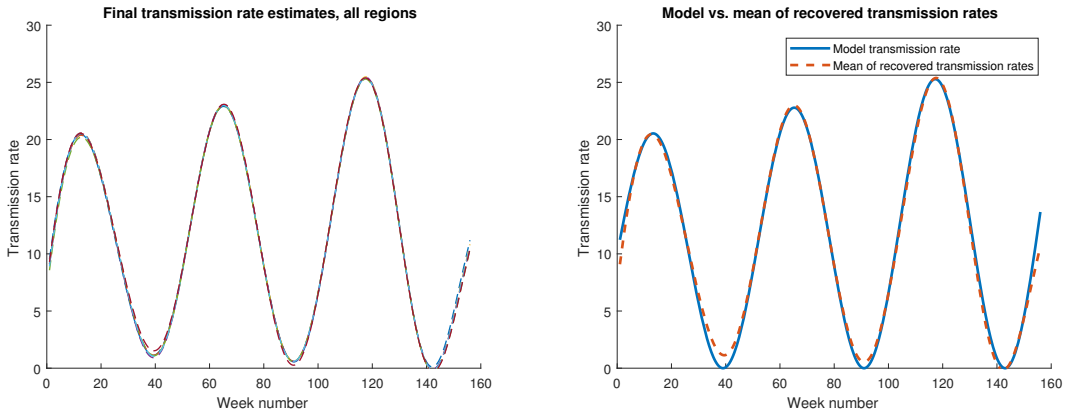

Figure (3.4) Left: Plot of the final transmission rate estimates across all regions. Right: Comparing model vs. mean of all transmission rates across all regions.

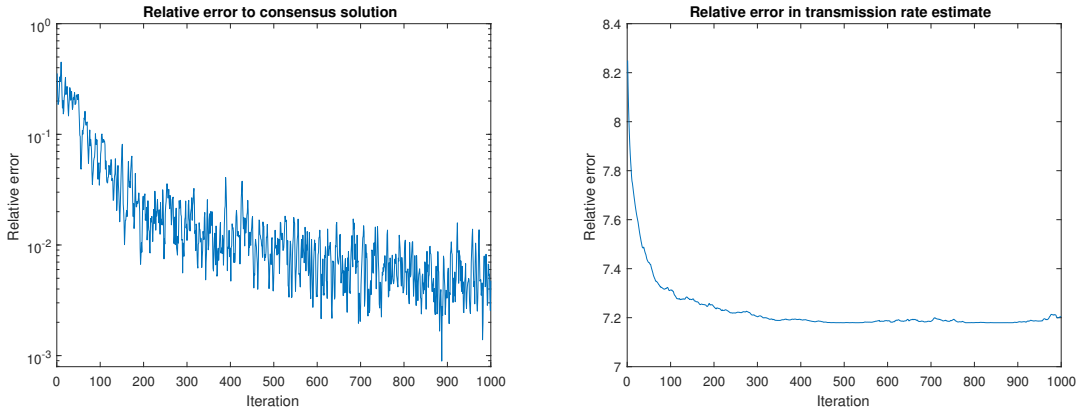

Figure (3.5) Plot of relative error decay in both consensus (left) and accuracy of transmission rate recovery (right).

We plot the final transmission rate estimates for all regions in Figure 3.4 next to the plot of their mean, compared to the model transmission rate. Furthermore, we plot relative error in consensus and relative error in estimated transmission rate in Figure 3.5. Finally, we compare the results of incidence case recovery across all 7 regions in Figure 3.6.

Confident that our method accurately estimates the transmission rate using full data, we turn to using partial data for transmission rate recovery. We use the same simulated incidence case data for each node, only this time we aim to recover the transmission rate using only part of this data. For this test, we set the partial data length to 52, taking 1 year of observed incidence cases as a priori knowledge. We wish to see how the recovered transmission rates

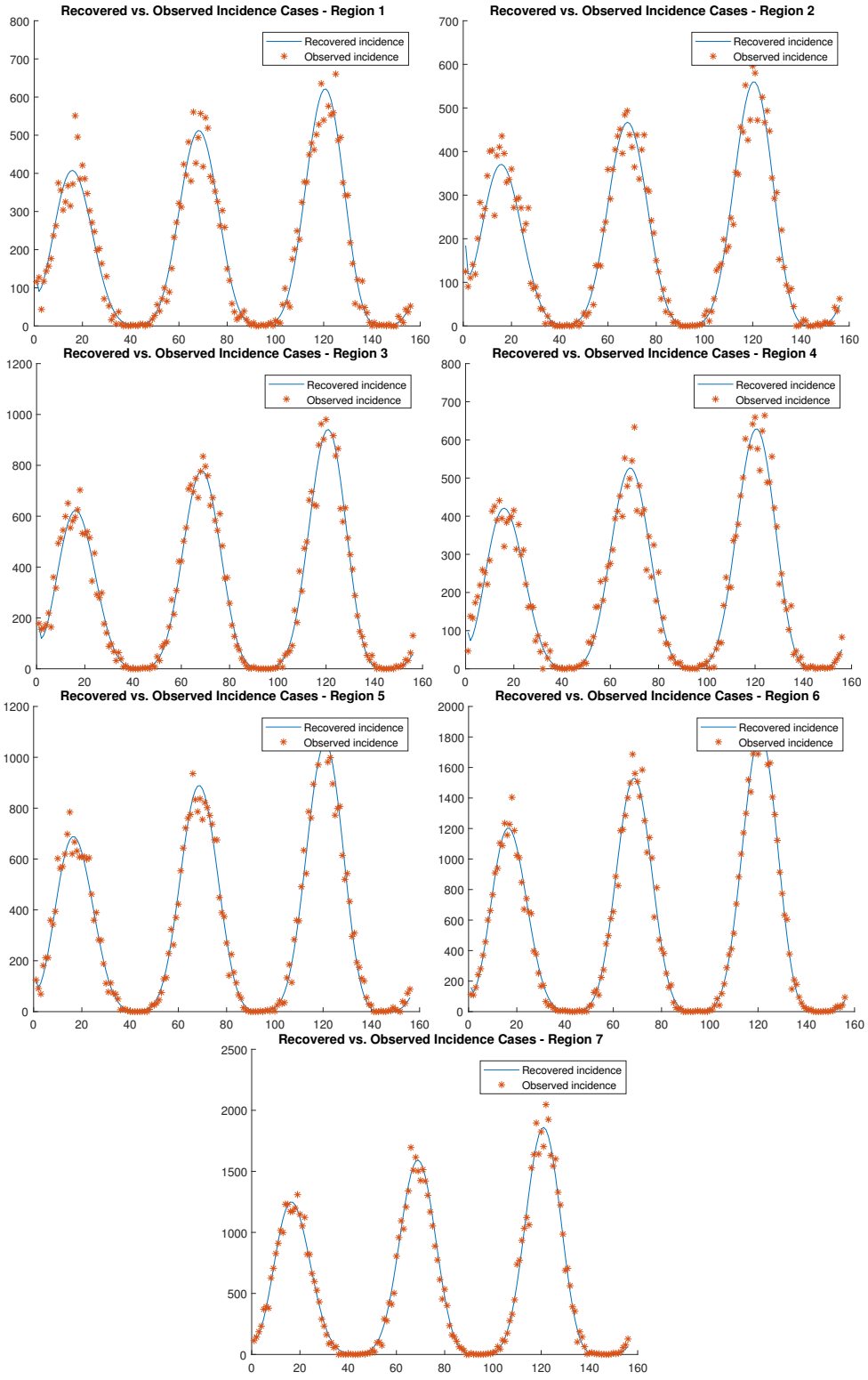

Figure (3.6) Comparing observed incidence with the incidence recovered by the method.

compare to the model transmission rate under this partial data assumption. We also wish to see whether the recovered incidence cases closely follow the patterns in the observed data with the goal of comparing the accuracy of incidence case forecasting, using the simulated incidence cases as a baseline. We use the same mixing matrix and step-size rule as for the full data case, but we make a minor modification to the basis functions to assist in coefficient vector recovery. When using partial data, as with the single node case in our previous work, the basis functions are lifted away from zero, which helps reduce the occurence of negative transmission rate values during coefficient vector recovery. Such values decrease the stability of the system and lead to less accurate results in the final transmission rate estimate and incidence case recovery. The modified basis used for this partial data scenario is of the same form $\{s_n(u(t)), c_n(u(t)) : 1 \leq N \leq 14\} \cup \{1\}$, though here we define $s_n(u(t)) = 0.15(\sin(2\pi n \cdot u(t)/L) + 4)$ and $c_n(u(t)) = 0.15(\cos(2\pi n \cdot u(t)/L) + 4)$, and $L = 4$. We choose $L = 4$ so that the recovered transmission rate exhibits cyclic spikes with period close to the 1 year period taken as a priori knowledge. The constant factors and terms modifying the basis functions are not required to take on these chosen values, and in general we only require the basis functions to be bound far enough away from zero as to minimize the occurence of negative transmission rate values when using the method to recover the coefficient vectors.

As before, we let each node randomly select an initial constant value uniformly, this time from the interval $[1.5, 2.5]$, with each node drawing its sample independently. Each node defines its preliminary coefficient vector $\hat{A}_i^0$ and uses this value to initialize the mini-loop, which we run for 20 iterations to produce the initial, non-constant estimate for β_0. This is used as before, to generate the initial coefficient vector A_i^0 at each node. The main loop is run for 250 iterations, after which each node takes its transmission rate estimate and performs incidence case recovery and projection. In this partial data case, we compare the accuracy of the recovered (past) incidence cases while also running a forecast on this recovered incidence case data vector. In Figure 3.7 we see a plot of the transmission rate estimates for all nodes, and we compare the mean transmission rate to the model transmission rate. We see that the recovered transmission rate estimates are very accurate up to the partial data cutoff.

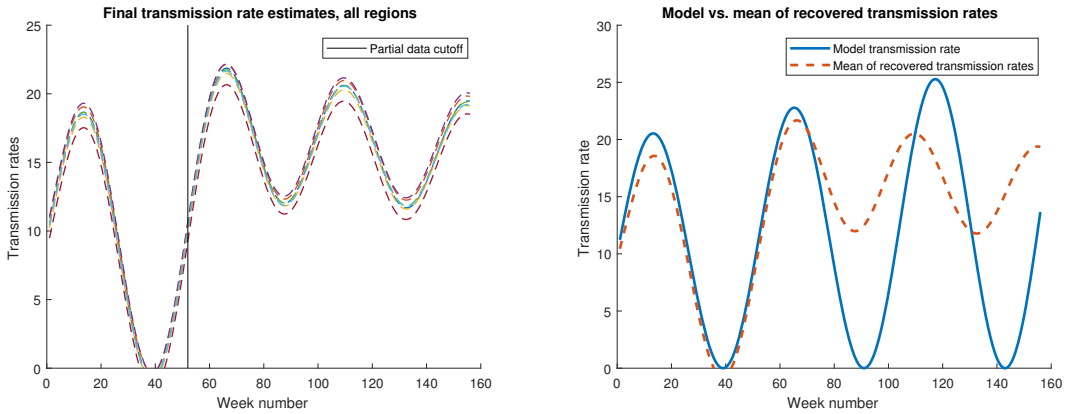

Figure (3.7) Plot of each node's transmission rate estimate (left) and a plot of the mean of all transmission rates compared to the model transmission rate (right).

After this point they no longer closely follow the model transmission rate, though exhibiting the general periodicity taken as a priori knowledge of the cycle length for this disease. In Figure 3.8 we see each node's recovered incidence case data, as well as each node's forecast incidence case data. As expected, the recovered incidence cases closely mirror the simulated outbreak data.

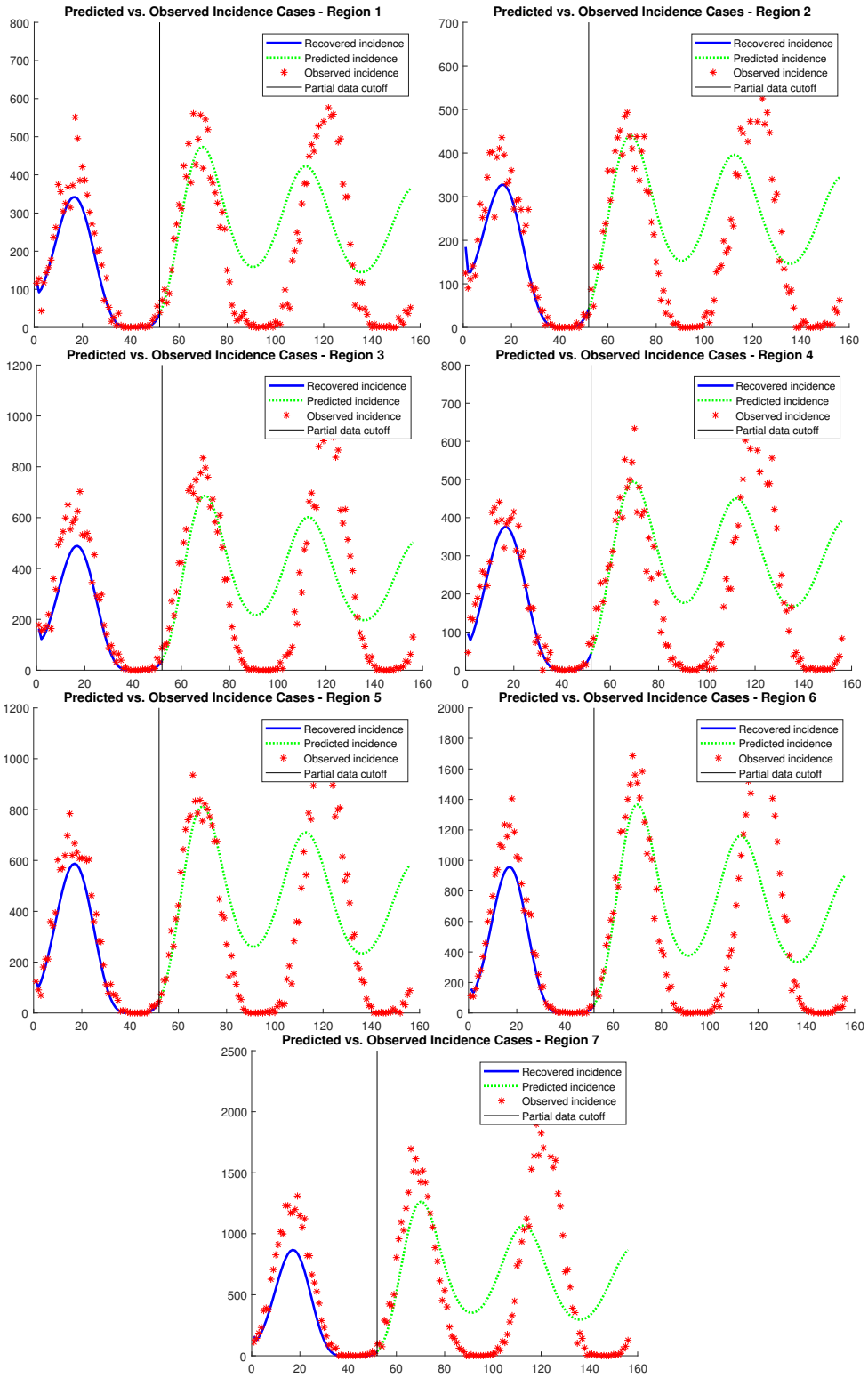

Figure (3.8) Comparing observed incidence with incidence recovered by the method using partial data.

REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[2] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *The Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[3] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan, "Mlbase: A distributed machine-learning system." in *CIDR*, vol. 1, 2013, pp. 2–1.

[4] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Advances in Neural Information Processing Systems*, 2014, pp. 19–27.

[5] F. Iutzeler, P. Ciblat, W. Hachem, and J. Jakubowicz, "New broadcast based distributed averaging algorithm over wireless sensor networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3117–3120.

[6] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 20–27.

[7] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 305–318.

[8] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *Power Systems, IEEE Transactions on*, vol. 28, no. 2, pp. 940–951, 2013.

[9] C.-H. Lo and N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 66–77, 2013.

[10] L. Zhao, W.-Z. Song, L. Shi, and X. Ye, "Decentralised seismic tomography computing in cyber-physical sensor systems," *Cyber-Physical Systems*, pp. 1–22, 2015.

[11] L. Zhao, W.-Z. Song, and X. Ye, "Fast decentralized gradient descent method and applications to in-situ seismic tomography," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 908–917.

[12] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 32–43, 2014.

[13] A. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[14] T.-H. Chang, W.-C. Liao, M. Hong, and X. Wang, "Asynchronous distributed admm for large-scale optimization part ii: Linear convergence analysis and numerical performance," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3131–3144, 2016.

[15] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed admm for large-scale optimization part i: Algorithm and linear convergence analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3118–3130, 2016.

[16] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar, "An asynchronous parallel stochastic coordinate descent algorithm," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 285–322, 2015.

[17] E. Wei and A. Ozdaglar, "On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 551–554.

[18] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1701–1709.

[19] D. Jakovetic, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1131–1146, 2014.

[20] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, 2009.

[21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[22] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[23] Y.-P. Tian and C.-L. Liu, "Consensus of multi-agent systems with diverse input and communication delays," *Automatic Control, IEEE Transactions on*, vol. 53, no. 9, pp. 2122–2128, 2008.

[24] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." DTIC Document, Tech. Rep., 1984.

[25] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus admm," *Signal Processing, IEEE Transactions on*, vol. 63, no. 2, pp. 482–497, 2015.

[26] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 892–904, 2016.

[27] D. Jakovetic, J. M. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented lagrangian algorithms," *Automatic Control, IEEE Transactions on*, vol. 60, no. 4, pp. 922–936, 2015.

[28] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed admm over networks," *IEEE Transactions on Automatic Control*, 2017.

[29] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *Signal Processing, IEEE Transactions on*, vol. 62, no. 7, pp. 1750–1761, 2014.

[30] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[31] A. Mokhtari and A. Ribeiro, "Decentralized double stochastic averaging gradient," in *Signals, Systems and Computers, 2015 49th Asilomar Conference on*. IEEE, 2015, pp. 406–410.

[32] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *Automatic control, IEEE Transactions on*, vol. 57, no. 3, pp. 592–606, 2012.

[33] D. Yuan, D. W. Ho, and S. Xu, "Regularized primal-dual subgradient method for distributed constrained optimization," *IEEE Transactions on Cybernetics*, 2015.

[34] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2748–2761, 2009.

[35] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *Automatic Control, IEEE Transactions on*, vol. 60, no. 3, pp. 601–615, 2015.

[36] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.

[37] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," in *Proceedings of IEEE Asilomar Conference on Signals, Systems, and Computers*, 2016.

[38] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[39] M. Li, D. G. Andersen, and A. Smola, "Distributed delayed proximal gradient methods," in *NIPS Workshop on Optimization for Machine Learning*, 2013.

[40] S. Sra, A. W. Yu, M. Li, and A. J. Smola, "Adadelay: Delay adaptive distributed stochastic convex optimization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, vol. 51, 2016, pp. 957–965.

[41] W. Zhang, S. Gupta, X. Lian, and J. Liu, "Staleness-aware async-sgd for distributed deep learning," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 2350–2356.

[42] O. Shamir and N. Srebro, "Distributed stochastic optimization and learning," in *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on.* IEEE, 2014, pp. 850–857.

[43] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "A delayed proximal gradient method with linear convergence rate," in *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on.* IEEE, 2014, pp. 1–6.

[44] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, 2015.

[45] H. Wang, X. Liao, T. Huang, and C. Li, "Cooperative distributed optimization in multiagent networks with delays," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, no. 2, pp. 363–369, 2015.

[46] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul Erdős is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[47] A. H. Sayed, S.-Y. Tu, and J. Chen, "Online learning and adaptation over networks: More information is not necessarily better," in *Information Theory and Applications Workshop (ITA), 2013*. IEEE, 2013, pp. 1–8.

[48] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[49] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o (1/k2)," Doklady AN SSSR, Tech. Rep. 3, 1983.

[50] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, 2008.

[51] F. Brauer and C. Castillo-Chavez, *Mathematical Models in Population Biology and Epidemiology*. Springer New York, 2012.

[52] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 115, no. 772, pp. 700–721, aug 1927.

[53] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, jan 2000.

[54] A. Elkadry, "Transmission rate in partial differential equation in epidemic models," Master's thesis, Marshall University, 2013.

[55] C. Kirkeby, T. Halasa, M. Gussmann, N. Toft, and K. Græsbøll, "Methods for estimating disease transmission rates: Evaluating the precision of poisson regression and two novel methods," *Scientific Reports*, vol. 7, no. 1, aug 2017.

[56] A. Mummert, "Studying the recovery procedure for the time-dependent transmission rate(s) in epidemic models," *Journal of Mathematical Biology*, vol. 67, no. 3, pp. 483–507, jun 2012.

[57] S. F. Dowell, "Seasonal variation in host susceptibility and cycles of certain infectious diseases," *Emerging Infectious Diseases*, vol. 7, no. 3, pp. 369–374, jun 2001.

[58] M. Pollicott, H. Wang, and H. Weiss, "Recovering the time-dependent transmission rate from infection data via solution of an inverse ode problem."

[59] P. E. M. FINE and J. A. CLARKSON, "Measles in england and wales—i: An analysis of factors underlying seasonal patterns," *International Journal of Epidemiology*, vol. 11, no. 1, pp. 5–14, 1982.

[60] L. DeCamp, "Regularized numerical algorithms for stable parameter estimation in epidemiology," Ph.D. dissertation, Georgia State University, 2017.

[61] M. Pollicott, H. Wang, and H. H. Weiss, "Extracting the time-dependent transmission rate from infection data via solution of an inverse ODE problem," *Journal of Biological Dynamics*, vol. 6, no. 2, pp. 509–523, mar 2012.

[62] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, vol. 19, no. 92, pp. 577–577, 1965.

[63] B. Kaltenbacher, "On broyden's method for the regularization of nonlinear ill-posed problems," *Numerical Functional Analysis and Optimization*, vol. 19, no. 7-8, pp. 807–833, jan 1998.

[64] A. Smirnova, "On TSVD regularization for a broyden-type algorithm," *Journal of Inverse and Ill-posed Problems*, vol. 0, no. 0, dec 2017.

[65] "Infectious disease data," https://ms.mcmaster.ca/~bolker/measdata.html, retrieved February 16, 2018. [Online]. Available: https://ms.mcmaster.ca/~bolker/measdata.html

[66] R. M. M. R. M. Anderson, "Vaccination against rubella and measles: Quantitative investigations of different policies," *Journal of Hygiene*, pp. 259–325, 1983.